BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi

Issue 125 | January 2023 | magpi.cc

The official Raspberry Pi magazine

RASPBERRY PI STARTER GUIDE

- QUICKSTART SETUP
- PYTHON PRIMER
- ELECTRONICS ESSENTIALS

PLUS COMMAND LINE BASICS!

BUILD AN AUTOMATED AQUARIUM

MAKE A MINIATURE MARS ROVER

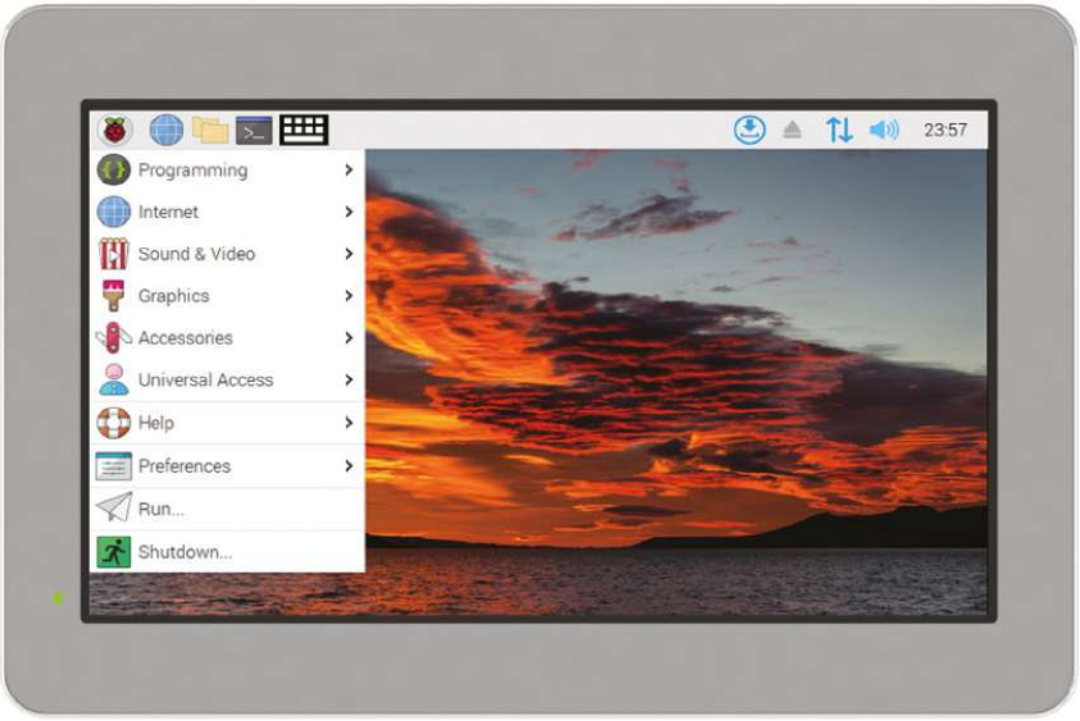
PUT A COMPUTER IN YOUR CAR

GLOBAL DELIVERY
magpi.cc/store

ART ATTACK! GET CREATIVE WITH RASPBERRY PI PICO

UL LISTED

Industrial Raspberry Pi

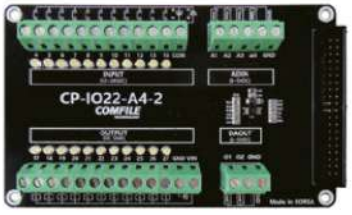


ComfilePi

The ComfilePi is a touch panel PC designed with high-tolerant components and no moving parts for industrial applications. It features a water-resistant front panel, touchscreen, color LCD (available in various sizes), RS-232, RS-485, Ethernet, USB, I2C, SPI, digital IO, battery-backed RTC (real-time clock), and piezo buzzer.

Use the rear-panel 40-pin GPIO header to expand its features and capabilities with additional I/O boards. The ComfilePi is UL Listed and employs Raspberry Pi Compute Module.

POWERED BY Raspberry Pi



Visit www.comfiletech.com

© copyright COMFILE Technology, Inc. ALL RIGHTS RESERVED

COMFILE TECHNOLOGY

WELCOME
to *The MagPi* 125

With any luck, you found a Raspberry Pi gift under the tree this Christmas.

If you're new: welcome to the party! Pull up a chair, pick up a copy of this magazine, and get ready for a wonderful world of making.

You'll find our Starter Guide on **page 28**, and a Raspberry Pi OS Guide on **page 36**. Newcomers to Linux should check out our Terminal Starter Guide (**page 48**).

Meanwhile, veterans looking for inspiration really should read our Arts & Crafts guide (**page 70**), and we have some wonderful tutorials this month: from putting a Raspberry Pi camera in an aquarium (**page 56**) to building a display kiosk (**page 60**).

I've loved putting together this issue. It's great to be reminded of just how exciting it is to pick up Raspberry Pi for the first time.

Happy Making!

Lucy Hattersley Editor

EDITOR
Lucy Hattersley

Editor of *The MagPi*. Lucy codes, crafts, and creates wonky robots. She's probably running in the Greenwich Pantomime Horse race about now!

magpi.cc

GET A RASPBERRY PI PICO W WITH A SUBSCRIPTION! PAGE 26

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
RASPBERRY PI
STARTER GUIDE
ART ATTACK! GET CREATIVE WITH RASPBERRY PI PICO

Explore E-ink technology with **EPDK** from PERVASIVE DISPLAYS

E-paper display (EPD) Pico Development Kit

The EPD Pico Development Kit from Pervasive Displays includes everything to start exploring E-ink technology.

⇒ EPD Extension board Gen 3 (EXT3-1) board
⇒ Raspberry Pi Pico RP2040 (with pin header)
⇒ 2.66" E-ink Monochrome Display with Fast Update
⇒ 20-pins and 2-pins bridging cable
⇒ Pin mapping label (3 pieces)

POWERED BY Raspberry Pi

ECO FRIENDLY

9/10
The MagPi 121

Grab a USB cable, download the library and start exploring E-ink technology today!

Learn more PERVASIVE DISPLAYS
www.pervasivedisplays.com

Buy it sparkfun ELECTRONICS
www.sparkfun.com/products/19219

Entaniya

Cube Case 01
for Single Board Computers

Cube Case 01 provides safe housing for main units of Single Board Computers, e.g. Raspberry Pi 3 or 4. It has a high-quality machined aluminum frame, with many options such as interchangeable acrylic panels and perforated panels that allow cables to pass through.

products.entaniya.co.jp/en/

Dustproof and drip-proof in use with the included sealing mats

A free configurable case with replaceable panels

Compatible with Entaniya All-Weather Case WC-01 for Raspberry Pi Camera Module 2 (sold separately)

For inquiries, please contact products@entaniya.co.jp

Entaniya Co.,Ltd., Nishiwaseda 2-5-13, Shinjuku-Ku, Tokyo 169-0051, Japan | www.entaniya.co.jp

Contents

Issue 125 > January 2023

Cover Feature

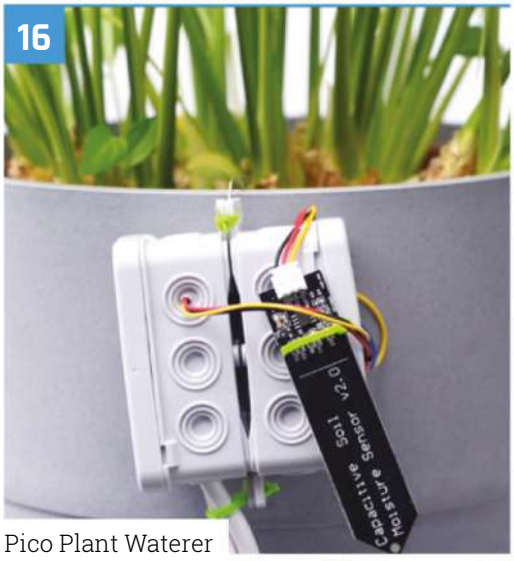
28 Raspberry Pi Starter Guide

Regulars

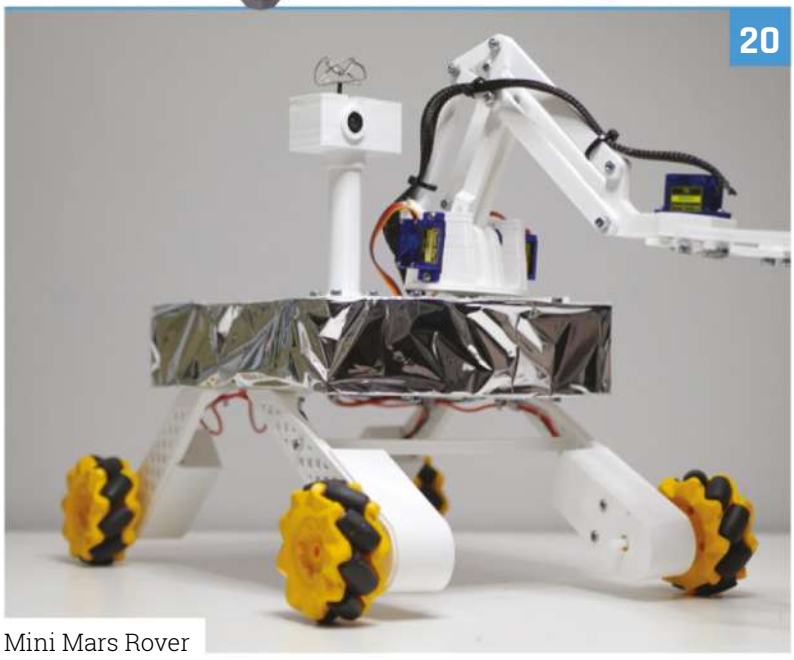
92 Your Letters
97 Next Month
98 The Final Word

Project Showcases

08 K-9
12 Maka Niu
16 Pico Plant Waterer
20 Mini Mars Rover
22 PiCar



Pico Plant Waterer



Mini Mars Rover

The MagPi magazine is published monthly by Raspberry Pi Ltd. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The MagPi magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Tutorials

48 Get started with command line
52 Build a DB9 to USB converter
56 Watch your aquarium anywhere
60 Using Raspberry Pi in kiosk mode
66 Build a Pico W web app

Pico DB9 to USB converter

Watch your aquarium

The Big Feature

Get creative with Raspberry Pi Pico

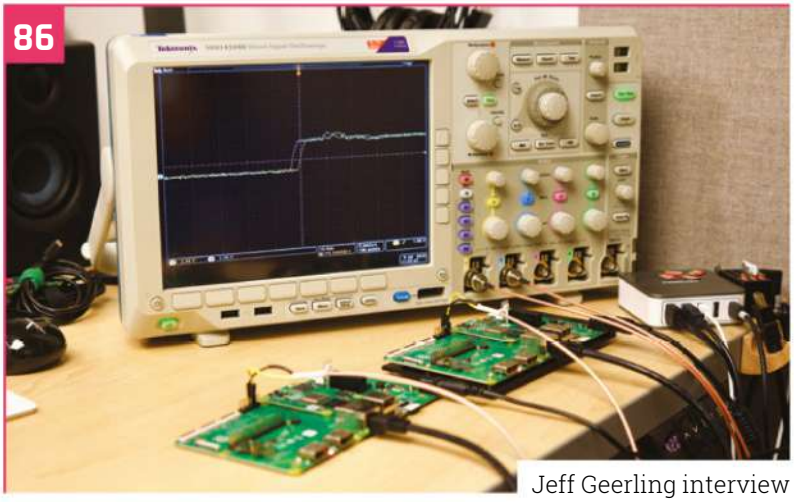
Top 10 Advanced projects

Reviews

78 ReTerminal Expansion E10-1
80 RelayFi
82 Top 10 Advanced projects
84 Learn web design

Community

86 Jeff Geerling interview
88 This Month in Raspberry Pi

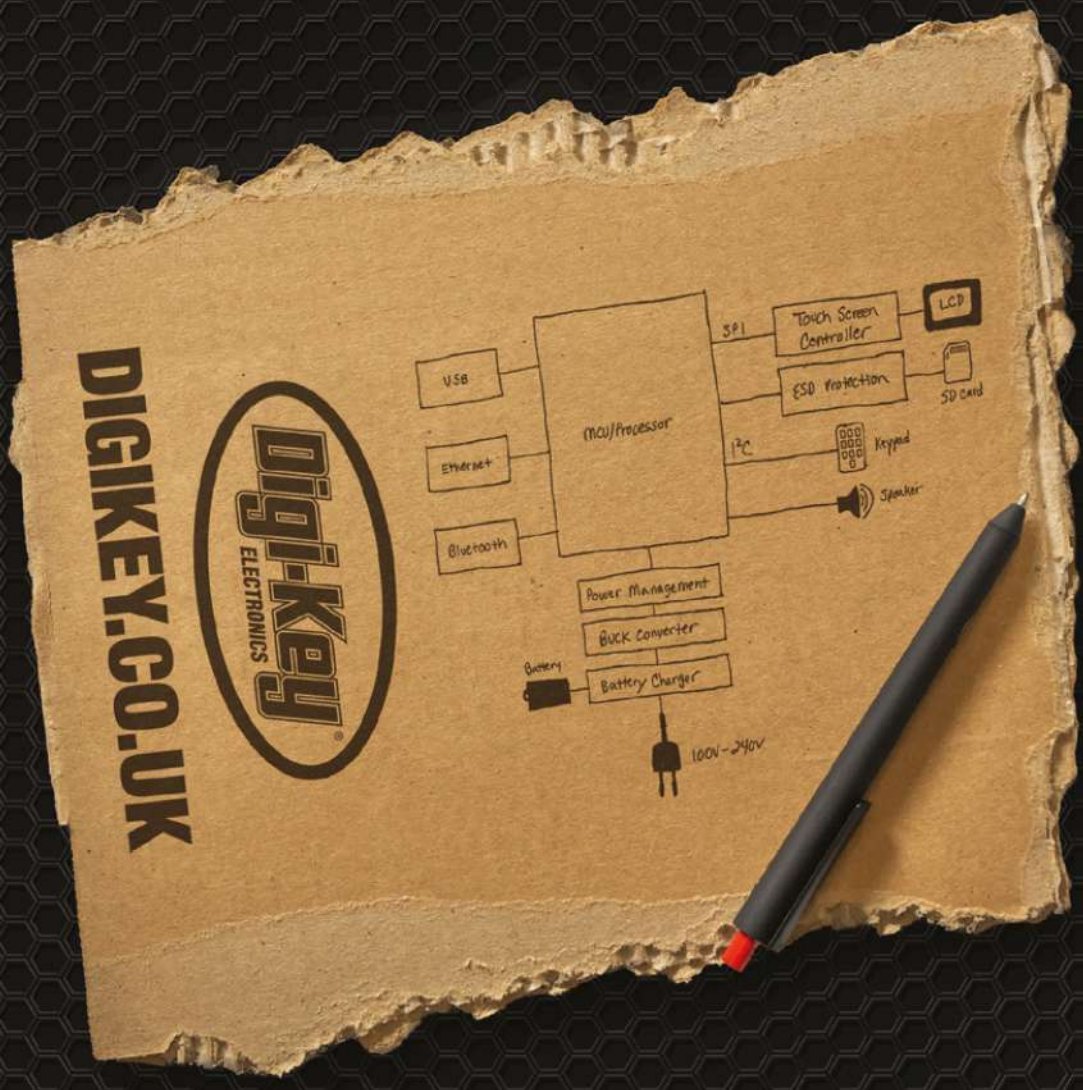


Jeff Geerling interview

WIN 1 OF 10 RASPBERRY PI **ZERO 2 W**

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

IDEAS
START HERE



From millions of in-stock parts to the latest new product inventory, we've got everything you need to turn breakthrough ideas into reality.

Get inspired at digikey.co.uk or call 0800 587 0991.

Digi-Key ELECTRONICS

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

K-9

Recreating a beloved TV character or prop is fairly exciting, **Rosie Hattersley** hears how Raspberry Pi allows one maker to give K-9 multiple Whovian features

MAKER

Fitz Walker

Fitz is a maker/tinkerer by nature with an unhealthy love for radio-controlled models of all types.

magpi.cc/hobbyview

Like many millions of us, "maker/tinkerer" Fitz Walker holds classic *Doctor Who* in very high regard. As an electrical engineer and computer scientist, he was better placed than many Whovians to show off his love of the show by building his own replica K-9, giving it "modern components" with Raspberry Pi.

Keeping the 'use whatever you've got to hand' ethos of the original *Doctor Who* sets, a loo roll holder disguises an unexpected K-9 defence mechanism.

The K-9 project helped Fitz fulfil an ambition to build a film replica of one of their "favourite companions to my favourite Doctor" [Tom Baker, naturally] and "to experiment with Raspberry Pi and other microcontrollers, and electronics to see how well they might replicate all the animatronic and visual features of the hero prop."

Fitz chose Raspberry Pi for its "powerful computer processing and video display", and options for interfacing with the physical world, giving K-9 "some level of autonomy and AI."

Design cues

Having been introduced to Raspberry Pi by a friend – who he thought was pranking him, based on the single-board computer's name – Fitz realised it would suit his needs very nicely. "Once I realised its potential, I kept Raspberry Pi in mind for any projects I engaged in." Raspberry Pi is "a powerful tool to do pretty much anything I can think of."

K-9 is the main focus of Fitz's tinkering efforts, and the only project for which he has had to do any custom design work, since some aspects could be adapted from other people's projects. In particular, founder of the K-9 Replica group Dave Everett had produced detailed templates and assembly guides for the animatronic dog's body, including a basic mechanical framework

The Tom Baker scarf is a nod to Fitz's favourite Doctor. He makes regular appearances on K-9's built-in screen displaying old TV clips

A Raspberry Pi 3 currently provides K-9's 'brains', helping him make tactical decisions and interact with humans and aliens

(**magpi.cc/k9bodyconstruction**), which were quickly seized on by fellow Whovian makers. Another impressive K-9 Raspberry Pi build that we've covered (**magpi.cc/k9prop**) uses this template too, or you can join a dedicated builder's Facebook group: **magpi.cc/k9facebook**.

Building on the original

Fitz also made use of existing designs for the very intriguing camera and AI element that he added to K-9's head. This is "based, in part, on the wedding photo booth (**magpi.cc/photoboothdiy**) by Anthony

K-9's head contains a camera. Photos are analysed by Raspberry Pi before K-9 prints out a pronouncement on whether a human or alien is in sight

Quick FACTS

- Fitz's K-9 extras include a loo roll holder nose…
- …which conceals a retractable laser gun
- There's a side monitor originally designed for cars…
- …which plays a random selection of *Dr Who* clips…
- …or sometimes random waveform sounds from YouTube

Sabatella which inspired K-9's photo-taking abilities," code which Fitz modified for K-9.

"Pretty much all the electronics and most of the animatronics were designed and set up by myself," says Fitz, who reports that the build was not complex but required "a bit of time and patience to build, program, and debug," with integrating all the parts the most challenging aspect. "Like any project, unexpected things come up. The drive system was a bit troublesome at first and had to be totally redone. Even then, it has some directional instability that needed to be rectified with a gyro."

▲ Elementary school pupils have been lucky enough to get a surprise visit too

▲ Fitz's K-9 has made various appearances at shows in the US, often flanked by Daleks

K-9 is a combination of 3D-printed parts and components sourced from online stores, including Adafruit and SparkFun. Since everything would eventually be hidden inside K-9's body, Fitz did all his prototyping on a separate bench using

> K-9 is a combination of 3D-printed parts and components sourced from online stores

a different Raspberry Pi, before transferring everything to the ready-assembled setup inside the robot. This requires some duplication of hardware, but makes life easier for a project with plenty of iterative revisions and "continuous modification and expansion." Utilising the various GPIO connections, to both read inputs from

Large bright buttons give K-9 an authentic 1970s look

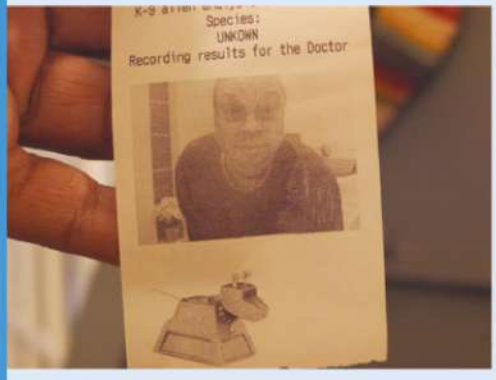
K-9's scarf shows his allegiance to his favourite Doctor, clips of whom might well appear on-screen imminently

an external trigger and to run various external devices, was a big challenge. He estimates the build took three months of fairly focused work, with some light additions over the course of the following year. After an outlay of roughly $1000, the only real running costs are for the thermal paper on which K-9 prints photos of whoever is in front of him, along with a wry take on whether they might be human or alien.

You can see Fitz's K-9 robot in action at **magpi.cc/k9chronicles**. Expect a new post soon, as he's currently working on adding voice-recognition and Pico elements.

Give K-9 new tricks

01 Download templates for K-9's body and an electronics schematic from **magpi.cc/k9bodyconstruction** to either 3D-print or laser-cut. You will need at least a Raspberry Pi 2, plus plenty of servos, and a fair amount of mechatronics knowledge.



02 If you want to add the AI camera feature, you can adapt code from this photo booth project: **magpi.cc/photoboothdiy**. You'll need a thermal printer to output K-9's 'human or alien' verdicts.

03 Fitz used Python on his Raspberry Pi 3 to control the camera/printer and eye lights in K-9's head, and to play video clips on K-9's side monitor.

Maka Niu

A deep-sea underwater imaging and sensing system made using a Raspberry Pi Zero and Camera Module is making a splash in marine exploration circles, **David Crookes** dives in

MAKER

Katy Croff Bell

Dr Katy Croff Bell is a deep-sea explorer with a background in ocean engineering, maritime archaeology, and geological oceanography. She is founder and president of the Ocean Discovery League.

magpi.cc/makaniu

It would seem there are few limits to where Raspberry Pi can go. The computer has been used in the Antarctic and on the International Space Station. It has also, as this project proves, plunged deep underwater – in this case, furthering scientific knowledge and enabling more people to explore the world's oceans.

Indeed, without Maka Niu – a small, low-cost, deep-sea image and data collection system that has Raspberry Pi at its heart – Katy Croff Bell, president of the Ocean Discovery League, says such exploration could continue to be the exclusive domain of the wealthy. "By making this technology open and available to anyone," she says, "we hope to bring more equity to this important field."

This has far-reaching benefits. Less than one percent of the deep sea-floor has been seen and characterised, while most coastal countries lack the tools needed. "Ultimately, more observations will result in more discoveries and a better understanding about the single largest biosphere on Earth," she adds.

Not bad for a project that was initially earmarked for the Festival of the Pacific Arts and Culture, aimed at teaching high school students about deep-sea exploration and research.

Sensing the ocean

"The idea for Maka Niu had been brewing for a few years while I led the Open Ocean Initiative at the MIT Media Lab," Katy explains. "The spark was a trip to Hawaii in January 2020 where my team and the MIT Future Ocean Lab were working with the Polynesian Voyaging Society and decided to create a small, low-cost camera."

The festival was cancelled due to Covid-19, but Katy says this allowed the group to take a step back and work on making the end product even better than originally envisioned. "In the last five to ten years, there has been a growing movement towards the development of lower-cost systems, but still many of them can't go very deep (tens or hundreds of metres) or they can only collect one type of data such as imagery," she says.

"Maka Niu can collect several types of data including video and still imagery, depth, temperature, and GPS location when out of the water. It can go up to 1500 metres below sea level, and we have designs to go 6000 metres which would enable it to reach 99 percent of the sea floor.

"It's designed to be modular so you can create other types of sensors or modules using the same housing, battery pack, and electronics. And we've designed the interface so that it's really easy to program, use, and see your video stills using a smartphone or tablet without internet access."

A deep mission

The use of Raspberry Pi felt natural. "The need for an imaging system that would be at least HD led to the adoption of Raspberry Pi Zero and Raspberry Pi Camera Module V2," affirms Ocean Discovery League imagineer Dan Novy. "Brennan Phillips, from the University of Rhode Island, had created an epoxy potted [Raspberry] Pi-based camera called DEEPi which showed promise, and the cost and physical dimension of Raspberry Pi Zero were perfect in keeping the device small but powerful."

Dan says there's a Node.js-based mission engine accessed over Wi-Fi from a browser on a phone, tablet, or laptop. "Each Maka Niu acts as its own

A Raspberry Pi Camera Module V2 fits neatly at the end of the project's housing

Raspberry Pi Zero is mounted inside Maka Niu and its on-board Wi-Fi allows communication and video download without opening the sealed, watertight unit

Calculations by Allan Adams, of Oceanic Labs, showed using polyoxymethylene for the main pressure housing would allow Maka Niu to descend at least 1500 metres

The project currently uses 10 mm acrylic for the main camera port, but using sapphire glass would allow the device to descend to 6000 metres, putting 99.9% of the ocean within imaging range

> More observations will result in more discoveries and a better understanding

access point for direct communication between the browser and the mission engine, or the device can join a Wi-Fi network in client mode and, again, become accessible via browser-to-program missions or to review and upload footage in the cloud or download it to local storage," he adds.

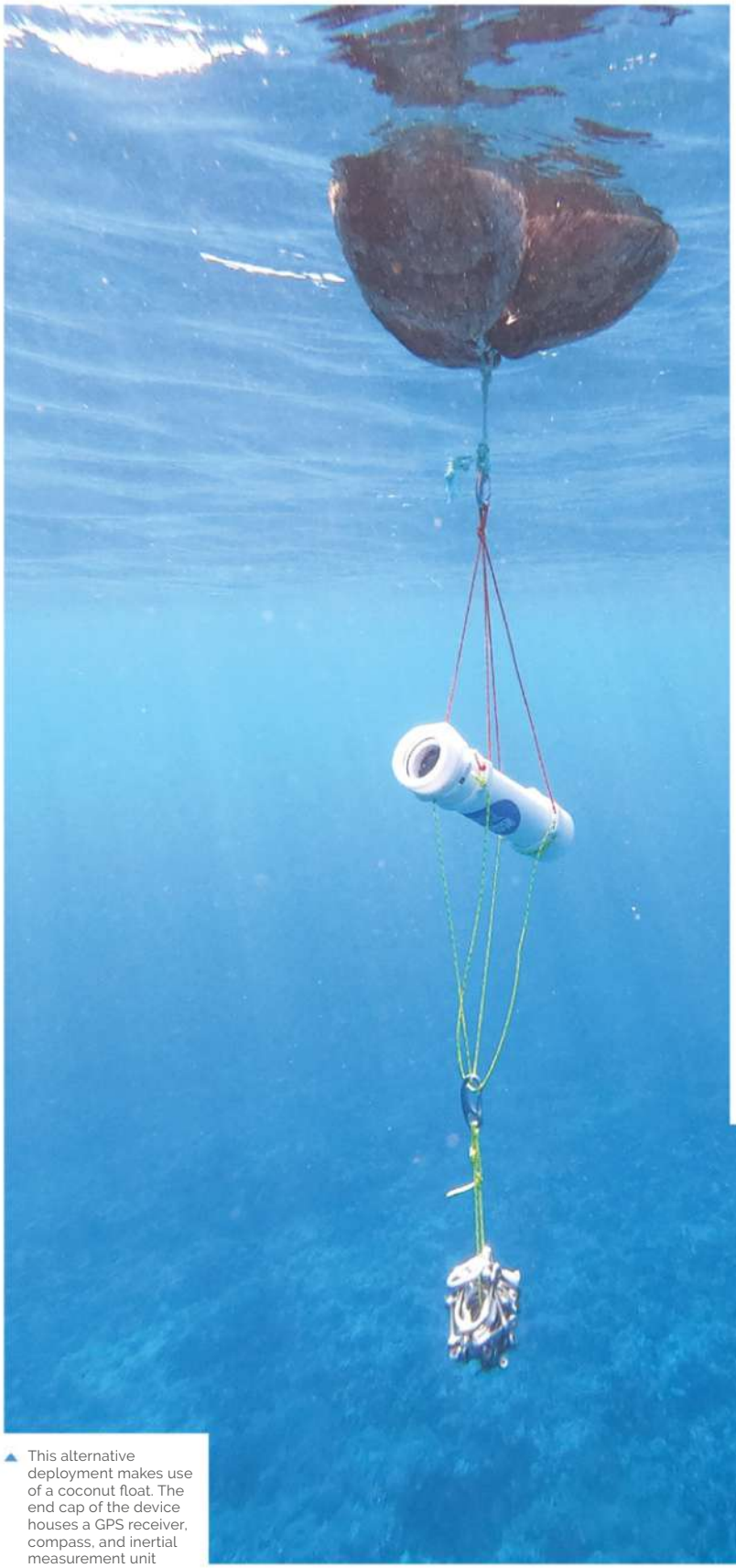
"The simplified programming environment allows the unit to be programmed in a LEGO-like fashion and allows non-programmers to be able to visually create missions just by stacking blocks of certain tasks and triggers into a loop that runs perpetually."

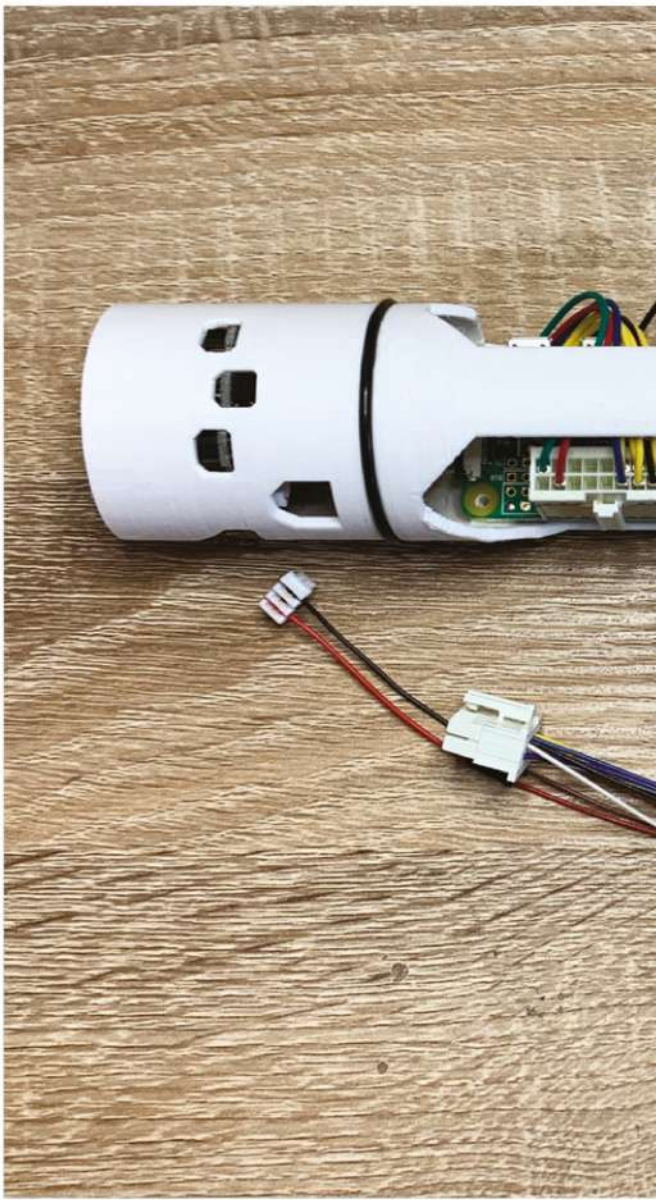
In the future, the idea is to use AI to automatically view and annotate mission footage. "We are working with CVision AI, scientists at the

Quick **FACTS**

- Maka Niu means "coconut eye"
- It has been designed to be modular
- The tech is open and available to all
- Data can be uploaded to image analyser Tator
- The ideal sensors were chosen by marine professionals

Using a pre-existing single-board computer such as Raspberry Pi Zero enables more people to contribute to Maka Niu's design and development





This alternative deployment makes use of a coconut float. The end cap of the device houses a GPS receiver, compass, and inertial measurement unit

▲ Dan Novy says inductive charging allows the device to be power-charged wirelessly

> The cost and physical dimension of Raspberry Pi Zero were perfect in keeping the device small but powerful

Monterey Bay Aquarium Research Institute, and others, on developing machine learning algorithms for automated detection and classification of organisms and other things," Katy explains. "The collaborative design is just one step along a long-term path of continued co-development."

Delving deep underwater

01 The imaging sensor is able to take videos and stills. This turtle image is very clear, although it was snapped in relatively shallow water.

02 There's a well-hidden rockfish in this image! All samples are time and location stamped down to the millisecond.

03 Maka Niu has captured a sea urchin. A small buzzer-style motor provides haptic feedback, letting the user know the device is taking an image or video.

Pico Plant Waterer

If the idea of asking a neighbour to water your houseplants while you're away is rather awks, this Raspberry Pi project will save your blushes, suggests **Rosie Hattersley**

MAKER

Vanessa Bradley and Martin Spendiff

VEEB began as a lockdown photography project for Vanessa Bradley and Martin Spendiff and soon morphed into a site documenting their builds.

veeb.ch

Sourdough baking, getting a pet and decluttering your home before filling it with houseplants to make it Instagram-worthy all had their moment during the Covid pandemic. Of these three trends, acquiring indoor plants appears to have had the most longevity. If you're off to visit friends and family over the festive break, or jumping on the long-vacation-to-avoid-winter-and-heating-bills bandwagon, it might be time to consider how your plants will fare while you're away. This Pico Plant Waterer saves you bothering a neighbour to pop in and do the honours.

For Vanessa Bradley and Martin Spendiff, collectively known as VEEB, the scenario came up because their "lovely" neighbour sometimes watered their houseplants, but the plants didn't always survive the experience. To save any more plants dying, they designed a carefully calibrated mechanism that would test whether the soil was dry enough for the plant to need watering, provide "a squirt" if needed, then check whether more moisture was required and apply it. Raspberry Pi was an obvious choice as they'd built a number of projects based on it before, some of which we've covered previously in *The MagPi*.

Vanessa originally worked in IT and Martin as a scientist, but they were "a bit bored" of their professions and became photographers instead. Their respective technical backgrounds and flair for visuals took off during lockdown, documenting their creations at **veeb.ch**. "We always had Raspberry Pi computers dotted about doing stuff. During lockdown we started building things that were a bit more involved," Martin explains.

He says one of the real pluses of using Raspberry Pi is documentation. They used a Pico for their plant watering project, which meant

VEEB's demo video celebrates a successful week of automatically watering, but not overwatering, their peace lily

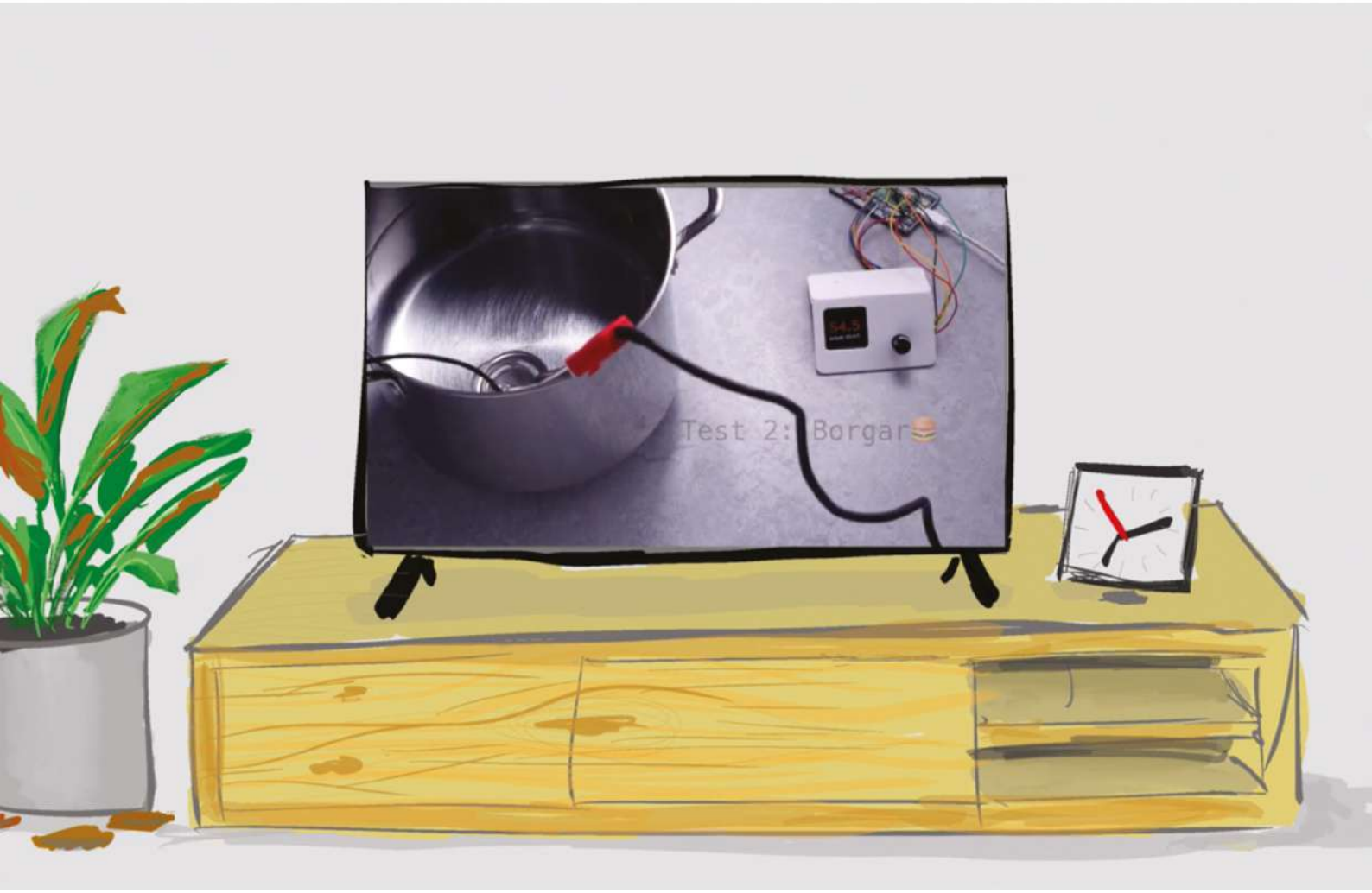
If the plant is thirsty, "a squirt" of water is syphoned into the plant pot, then the moisture levels checked again before any more watering is done

Raspberry Pi Pico uses information from a capacitive soil monitor pushed into the soil to determine whether the plant needs watering

▲ Martin and Vanessa's helpful neighbour "invariably kills the plants" she's been asked to water, "so the idea was to see if we could make something to replace her"

Quick **FACTS**

- Martin envisaged Clive Moss the Window Box Boss with...
- ...a blonde wig to look like their neighbour
- "We totally chickened out in case she ever saw it!"
- Clive originally had a screen and controller
- VEEB boxed him up instead for simplicity and cost



▲ Pico Plant Waterer is adapted from VEEB's automatic burger cooking project

losing some of the Python libraries in the switch to MicroPython, "but there isn't a huge leap from [Raspberry Pi] Zero to Pico and there is an elegant simplicity to a little board that just does one thing, the moment that you turn it on."

If it ain't broke...

VEEB projects tend towards the weird and slightly wacky, such as an old Teasmade linked to a Google Calendar (**magpi.cc/goblineteasmade**) and a Boombox Jukebox (**magpi.cc/boomboxjukebox**). "The general pattern that has emerged is taking some simple/older technology, linking it to newer tools, figuring out something that could be useful, making a simple proof of concept," says Martin.

Clive Moss the Window Box Boss, as the Pico Plant Waterer has been named, is really a revamp of a VEEB project which cooks burgers in a water bath (**magpi.cc/betterburger**). "It's a pretty general-purpose controller that can be applied to various situations that require some sort of monitoring/intervention." Aside from the switch to Pico, little hardware was needed beyond what they had lying around. Martin and Vanessa bought a cheap fish tank pump and got any extras they needed from the local hardware store.

"We took the burger cooker, removed the screen and controls, and popped it all in a waterproof box. The only part we made was bending a paper clip to hang it on the side of a plant pot."

Intelligent watering system

Watering a plant is only half the equation; the other part is checking whether the soil is sufficiently wet. As Vanessa notes, "not all types of plant require the same level of moisture."

Warning!
Water & Relay

Water can damage low-voltage electrical equipment. Water with mains-voltage electricity can be deadly.

magpi.cc/electricalsafety

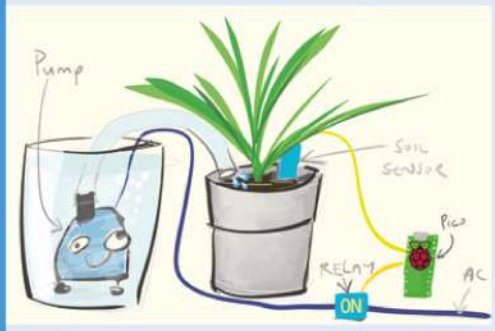
Clive Moss the Window Box Boss is "the man with the plan for a hosepipe ban"

Although the moisture sensor helps provide this information, common sense is needed too: Martin admits the first time they trialled the setup, all the water was pumped into the plant pot at once.

> We always had Raspberry Pi computers dotted about doing stuff

"We were too stupid to remember that the siphon effect was a thing, so on the first run of the code we just looked puzzled as a couple of gallons of water emptied into a plant pot while the pump wasn't even turned on." Moving the water tank further away helped. "The distance between water source and sensor means that when it has dried out, it drops a lot of water at once. This was initially an accident, but it turns out it promotes root growth."

Power up your plants



01 Aside from a Raspberry Pi Pico and a capacitive moisture sensor, you will need AA batteries, a pump, a large water container, a relay switch, and some clear tubing.

02 You can download the MicroPython setup code from **magpi.cc/sploosh** or view the demo video at **magpi.cc/youwatertheplant**. Once set up, fasten the tube to the edge of the plant pot.

03 The water source needs to be kept some distance from the plant to ensure it doesn't just decant the jug's contents straight into the plant pot. If all goes well, you should have a healthy plant, even after a week away from home.

Mini Mars Rover

Meet the inventive maker who decided to create his very own miniature Mars rover. **Nicola King** learns all about roving robotics

MAKER

Nikodem Bartnik

A 23-year-old maker from Poland, Nikodem is currently studying at the Silesian University of Technology. His projects are focused on robotics, programming, 3D printing, and CNC machines.

magpi.cc/nikodemyt

Attracted by Raspberry Pi Pico W's built-in Wi-Fi and the ability to program it in MicroPython, Nikodem Bartnik knew he had to make a project with one, but wasn't sure exactly what. Inspiration struck when he attended a CubeSat Summer School in Belgium. On a visit to the Euro Space Center, he saw a field with mini Mars rovers.

Nikodem had to fit the building of his own remote-controlled Mini Mars Rover around his other university work, and there was plenty to do, including designing it from scratch in CAD and undertaking a lot of 3D printing and CNC machining.

"All parts were designed in such a way that you can print them on the most popular FDM 3D printers and you don't need a huge build plate," he says. "The bottom plate was machined on my DIY CNC machine – IndyMill (**magpi.cc/indymill**)... The principle of how it works is very simple; designing and 3D printing it is not."

He also kept adding more features while working on the project. In particular, the robotic arm was an especially time-consuming part, with Nikodem designing and then redesigning it multiple times until it worked. The 3D printing aspect took around 24 hours in total but, in the end, all parts worked beautifully.

Core components

Raspberry Pico W is integral to the Rover: "Thanks to Pico W, it is possible to control it through Wi-Fi and [a] simple Python script on a computer," explains Nikodem. The script is created with Pygame and a simple custom class he developed for the simple user interface elements he needed.

"For controlling the micro servo motors, I am simply using 50Hz PWM signal. For DC motors I am using popular and cheap drivers, L298N, and control them through PWM and digital pins," he explains. He also added an FPV camera system to the build, which turned out to be far simpler than he'd imagined. "I had no experience with FPV systems before; it was completely new to me, but it turned out to be super-straightforward and it works great."

Small and sharable

Nikodem was determined to make his project as clean and small as possible "for aesthetics reasons, but also for its safety and functionality," he underlines. He also wanted to be able to share his work with others and make it easy to replicate, so he chose to make his own custom PCB, using a cheap CNC machine he'd modified. "The best part about this setup is: in the morning I design a PCB and, by the afternoon, I have it ready for soldering."

Nikodem's Rover has received plenty of positive comments and he admits "that was a big relief after working on it for so long." There were a few people who suggested that the Rover's wheels wouldn't survive the Mars environment, but Nikodem is pragmatic as "it was quite obvious for me from the beginning that the purpose of the project is to educate and inspire. Building it in the way real Mars rovers are built is completely unrealistic and unnecessary."

Excitingly, Nikodem is now concentrating on his bachelor thesis project which will use Pico's

The chassis parts were 3D-printed, including the mount for the first-person view camera

A robotic arm enables the Rover to pick up samples from the planet's surface

The Rover's mecanum wheels enable it to move sideways for maximum manoeuvrability

Quick **FACTS**

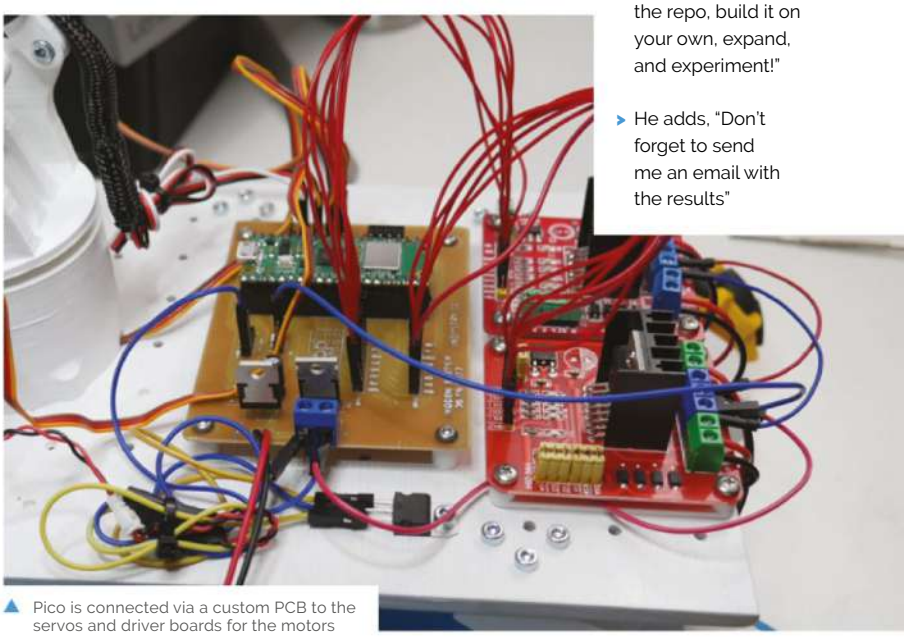
- Nikodem estimates the project cost him around $150 to make
- The Rover is open-source and all files are on GitHub: **magpi.cc/picomarsrover**
- It's powered by a 650 mAh 2S LiPo, with two voltage regulators on the custom PCB
- Nikodem says, "Feel free to fork the repo, build it on your own, expand, and experiment!"
- He adds, "Don't forget to send me an email with the results"

RP2040 chip. "I am currently testing the PCB for the project... The end goal is to be able to take long exposures of deep sky objects with a camera

> Thanks to Pico W, it is possible to control it through Wi-Fi and a simple Python script on a computer

by compensating the movement of Earth and also create a go-to system with Stellarium."

If you take a look at Nikodem's YouTube channel (**magpi.cc/nikodemyt**), you'll see a huge and interesting array of projects he's worked on so far, and it seems there will be plenty more to come from this very talented maker.



Pico is connected via a custom PCB to the servos and driver boards for the motors

PiCar

Upgrading a vehicle's entertainment system can be as simple or complex as you'd like. **Rob Zwetsloot** checks out a complex one

MAKER

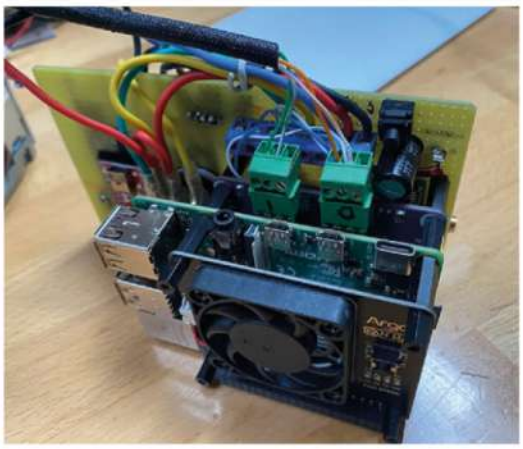
Vincent Moscaritolo

Retired software engineer specialising in privacy and cryptography, and one of the original Cypherpunks.

magpi.cc/picar

Tinkering and modifying a car is a time-honoured tradition among mechanics and engineers. Upgrading the sound system has always been a part of that too. However, that usually means putting in a pre-made head unit and maybe replacing the speakers. Vincent Moscaritolo has gone a step further with the sound system, giving it a full Raspberry Pi makeover.

"I have made extensive modification to my Jeep, including replacing the motor with a Chevy LS 6.0 V8," Vincent tells us. "This required some hacking into the Jeep's CAN bus to fool the Jeep into thinking it still has a motor. There is so much useful information present on the network that is not presented to the driver, and it frustrated me that one of the main user interfaces on the vehicle, the car radio, does not present or allow the driver to access it."



The full hardware powering the display behind the scenes

Vincent looked at various solutions, but they didn't quite fit the Jeep's aesthetic. He decided to make something a bit more custom using a VFD (vacuum fluorescent display) to fit the original look, and even fool folks into thinking it was the stock radio.

Farm to auto

After using Raspberry Pi to fix an automated chicken coop, Vincent started using it more in projects.

"I found that it was fairly easy to use Raspberry Pi to interface to the CAN bus as well as I2C," Vincent explains. "Since it was a Linux-based platform, it was also easy to talk to the various software-defined radio (RTL-SDR) devices out there."

Even with a familiarity with Raspberry Pi, and ease of access to the interfaces, the project hasn't been without issues.

"It has been a learning experience," Vincent says. "While there are a lot of folks building things for Raspberry Pi, not a lot of the software or hardware is rugged enough to put in an automotive application. A lot of tools, while pretty cool, were written at the hobbyist level. For instance, most of the software-defined radio (SDR) tools require a desktop display, and can't be used in a car radio app.

"The other problem was that the automotive manufacturers are really proprietary and secretive about how their systems work. I had to mostly adjust the Jeep CAN network packet information by trial and error, such as monitor the packets, open a car door, and see what changes, etc."

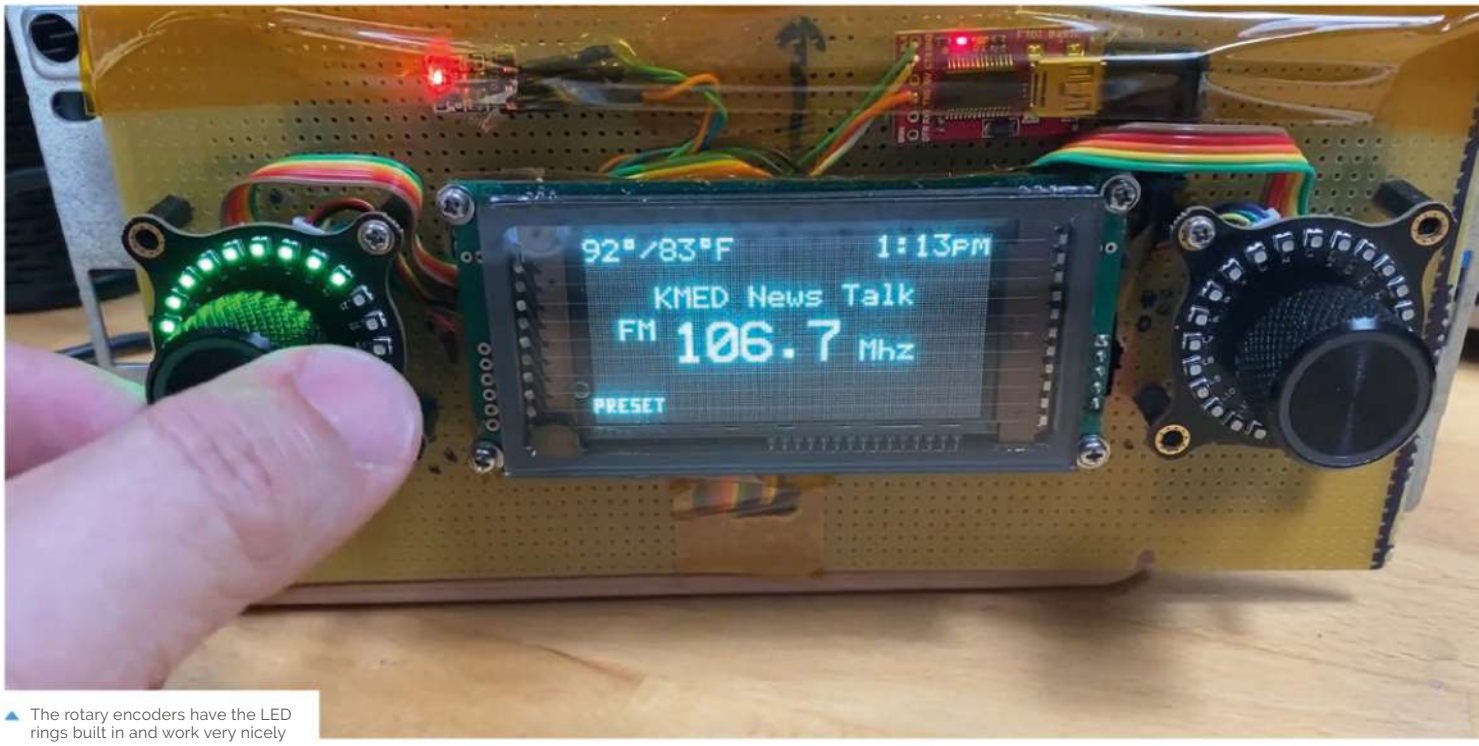
The screen replicates the original Jeep Wrangler stereo system
A little bit of painting and detailing is all that's left before it's installed
The rotary encoders have attached LED rings to show how much you've turned them
70°/83°F 1:31PM
So. County Fire
VHF 154.310 Mhz
-55
PRESET

Quick **FACTS**

- Vincent has been working on and off for a year
- The major functions have been working for six months
- OLED displays looked washed out in sunlight to Vincent
- Vincent moved the GPIO head to below the board
- It could run on a slower Raspberry Pi – it currently tops out at about 30% CPU

▲ AirPlay from Apple devices works, so people don't need to grab an aux cable

The GPS is more for positioning, which helps with hiking and such



The rotary encoders have the LED rings built in and work very nicely

▲ The display and knobs are to replicate the original stereo system on the Jeep

In the end, Vincent wrote a custom application in C++ that talks to the VFD over the serial port. It monitors the states of the knobs and the CAN bus, talks to a GPS board via serial, and the SDR via USB.

> I found that it was fairly easy to use Raspberry Pi to interface to the CAN bus as well as I2C

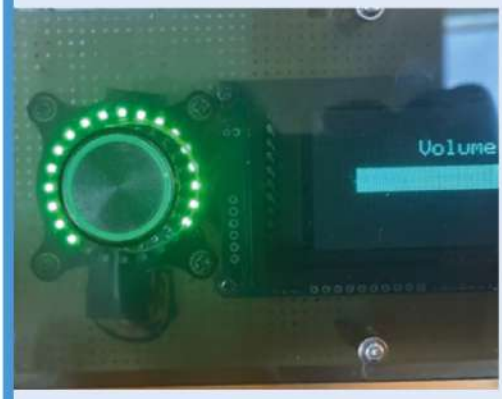
Finishing touches

"[It's] just about ready to replace the existing radio," Vincent tells us. "I am currently alternating between it sitting on my shop desk and my dashboard. I have a temporary harness cabled that allows me to plug it [in] and move it back and forth. I need to paint and engrave the faceplate."

Apparently reactions have been varied – folks who know what he's made are blown away while others are at least appreciative of the display. Count us in as one of the blown-away parties.

How it works

01 Data is taken from the CAN (Controller Area Network), a standard car data port that allows access to various information that mechanics and sometimes even the car can read. This can include engine operations, tyre pressure, fuel supply, etc.



02 With some hacking, the data is stored and translated to work on the limited VFD via several predetermined screens. The code also looks out for changes in the knobs to know when to display it.

03 The head unit displays regular information such as the radio, but can be changed for more detailed engine statistics, or even GPS, using the rotary dials like normal knobs. It also maintains the standard Jeep radio aesthetic

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W

+

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- Free delivery to your door
- £10 (UK only)

Subscribe for 6 Months

- Free Pico W
- 6 issues of The MagPi
- Free delivery to your door

£30 (UK) £35 (USA)
£35 (EU) £45 (Rest of World)

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a 6-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

WORTH $6

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
RASPBERRY PI
STARTER GUIDE
BUILD AN AUTOMATED AQUARIUM
MAKE A MINIATURE MARS ROVER
PUT A COMPUTER IN YOUR CAR
ART ATTACK! GET CREATIVE WITH RASPBERRY PI PICO

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

RASPBERRY PI STARTER GUIDE

SETTING UP RASPBERRY PI IS PRETTY STRAIGHTFORWARD, JUST FOLLOW THE ADVICE OF **ROSIE HATTERSLEY**

Congratulations on becoming a Raspberry Pi explorer. We're sure you'll enjoy discovering a whole new world of computing and the chance to handcraft your own games, control your own robots and machines, and share your experiences with other Raspberry Pi fans.

Getting started won't take long: just corral the extra bits and bobs you need on our checklist. Useful additions include some headphones or speakers if you're keen on using Raspberry Pi as a media centre, or a gamepad for use as a retro games console.

To get set up, simply use your pre-written microSD card (or use Raspberry Pi Imager to set up a card) and connect all the cables. This guide will lead you through each step. You'll find the Raspberry Pi OS, including coding programs and office software, all available to use. After that, the world of digital making with Raspberry Pi awaits you.

What you need

All the bits and bobs you need to set up a Raspberry Pi computer

A Raspberry Pi

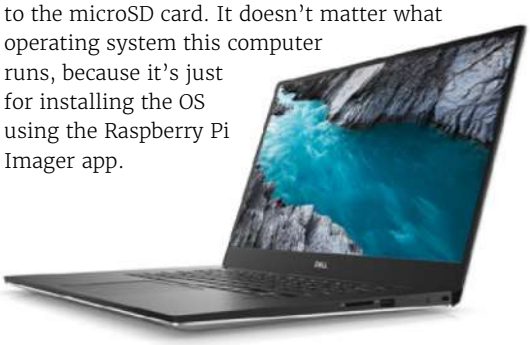
Whether you choose the new Raspberry Pi 400; or a Raspberry Pi 4, 3B+, 3B; Raspberry Pi Zero or Zero 2 (or an older model of Raspberry Pi), basic setup is the same. All Raspberry Pi computers run from a microSD card, use a USB power supply, and feature the same operating systems, programs, and games.

8GB microSD card

You'll need a microSD card with a capacity of 8GB or greater. Your Raspberry Pi uses it to store games, programs, and boot the operating system. Many Raspberry Pi computer kits come with a card pre-written with Raspberry Pi OS. If you want to reuse an old card, you'll need a card reader: either USB or a microSD to full-sized SD (pictured).

Windows/Linux PC or Mac computer

You'll need a computer to write Raspberry Pi OS to the microSD card. It doesn't matter what operating system this computer runs, because it's just for installing the OS using the Raspberry Pi Imager app.



USB keyboard

Like any computer, you need a means to enter web addresses, type commands, and otherwise control Raspberry Pi. The Raspberry Pi 400 comes with its own keyboard. Raspberry Pi sells an official Keyboard and Hub (**magpi.cc/keyboard**) for other models.

USB mouse

A tethered mouse that physically attaches to your Raspberry Pi via a USB port is simplest and, unlike a Bluetooth version, is less likely to get lost just when you need it. Like the keyboard, we think it's best to perform the setup with a wired mouse. Raspberry Pi sells an Official Mouse (**magpi.cc/mouse**).

Power supply

Raspberry Pi 4 and Raspberry Pi 400 need a USB Type-C power supply. Raspberry Pi sells power supplies (**magpi.cc/usbcpower**), which provide a reliable source of power. Raspberry Pi 1, 2, 3, and Zero models need a micro USB power supply (**magpi.cc/universalpower**).

Display and HDMI cable

A standard PC monitor is ideal, as the screen will be large enough to read comfortably. It needs to have an HDMI connection, as that's what's fitted on your Raspberry Pi board. Raspberry Pi 4 and 400 can power two HDMI displays, but require a micro-HDMI to HDMI cable. Raspberry Pi 3B+ and 3A+ both use regular HDMI cables; Raspberry Pi Zero W needs a mini-HDMI to HDMI cable (or adapter).

USB hub

Raspberry Pi Zero and Model A boards have a single USB socket. To attach a keyboard and mouse (and other items), you should get a four-port USB hub (or use the official USB Keyboard and Hub which includes three ports). Instead of standard-size USB ports, Raspberry Pi Zero has a micro USB port (and usually comes with a micro USB to USB-A adapter).

The USB-C socket is used to connect power to the Raspberry Pi 400. You can use a compatible USB-C power adapter (found on recent mobile phones) or use a bespoke power adapter such as the Raspberry Pi 15.3W USB-C Power Supply
The Ethernet socket can be used to connect Raspberry Pi 400 directly to a network router (such as a modem/router at home) and get internet access. Alternatively, you can choose a wireless LAN network during the welcome process

SET UP RASPBERRY PI 400

Raspberry Pi 400 has its own keyboard – all you need to connect is the mouse and power

01 Connect a mouse

Connect a wired USB mouse to the white USB connection on the rear of Raspberry Pi 400. The two blue USB 3.0 connectors are faster and best reserved for external drives and other equipment that need the speed.

02 Attach the micro-HDMI cable

Next, connect a micro-HDMI cable to one of the micro-HDMI sockets on the rear of Raspberry Pi 400. The one next to the microSD card slot is the first one, but either connection should work. Connect the other end of the HDMI cable to an HDMI monitor or television.

03 The microSD

If you purchased a Raspberry Pi 400 Personal Computer Kit, the microSD card will come with Raspberry Pi OS pre-installed. All you need to do is connect the power and follow the welcome instructions. If you do not have a Raspberry Pi OS pre-installed microSD card, follow the instructions later in 'Set up the software'.

SET UP RASPBERRY PI

Raspberry Pi 4 / 3B+ / 3 has plenty of connections, making it easy to set up

01 Hook up the keyboard

Connect a regular wired PC (or Mac) keyboard to one of the four larger USB-A sockets on a Raspberry Pi 4 / 3B+/ 3. It doesn't matter which USB-A socket you connect it to. It is possible to connect a Bluetooth keyboard, but it's much better to use a wired keyboard to start with.

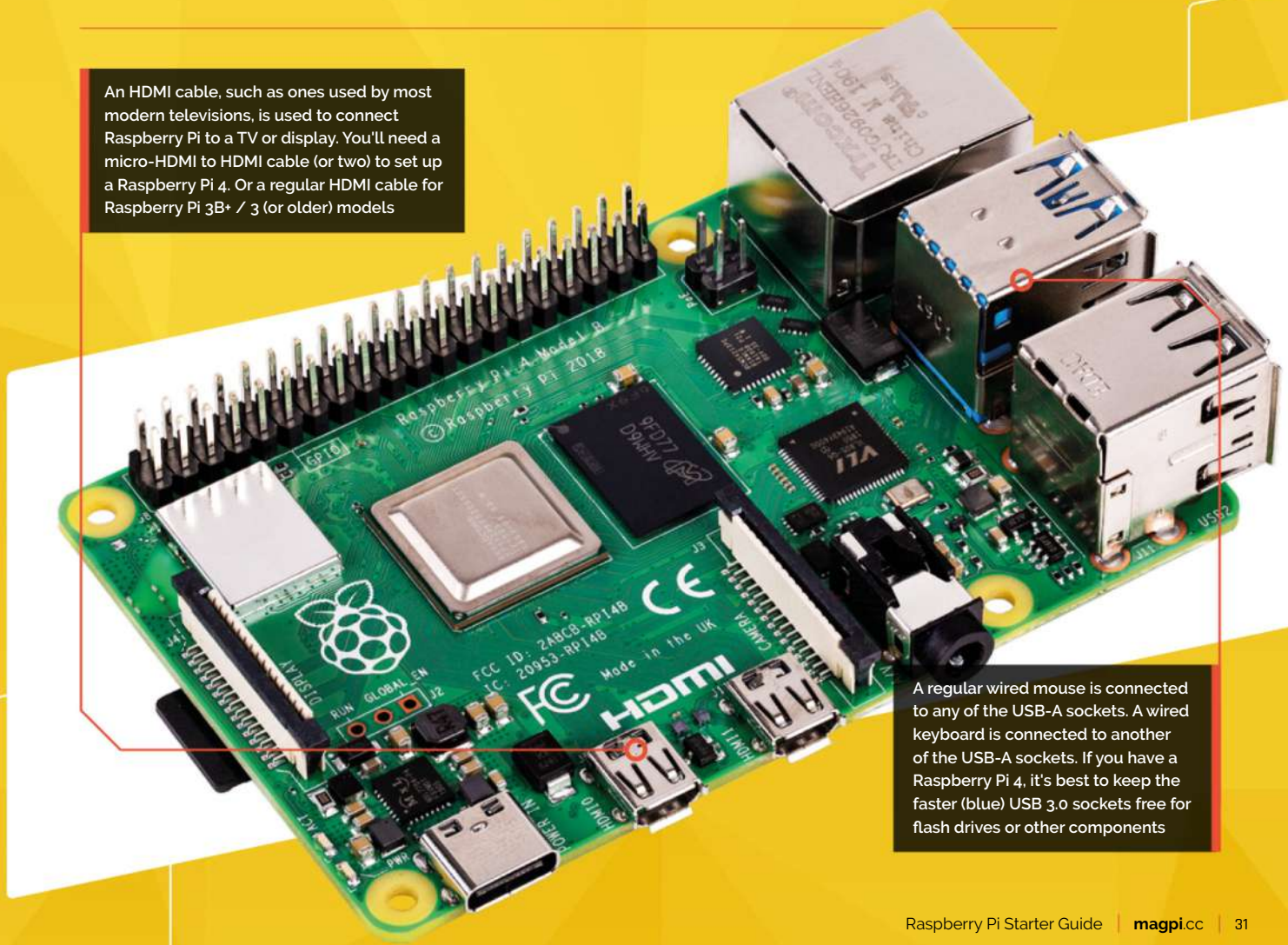
02 Connect a mouse

Connect a USB wired mouse to one of the other larger USB-A sockets on Raspberry Pi. As with the keyboard, it is possible to use a Bluetooth wireless mouse, but setup is much easier with a wired connection.

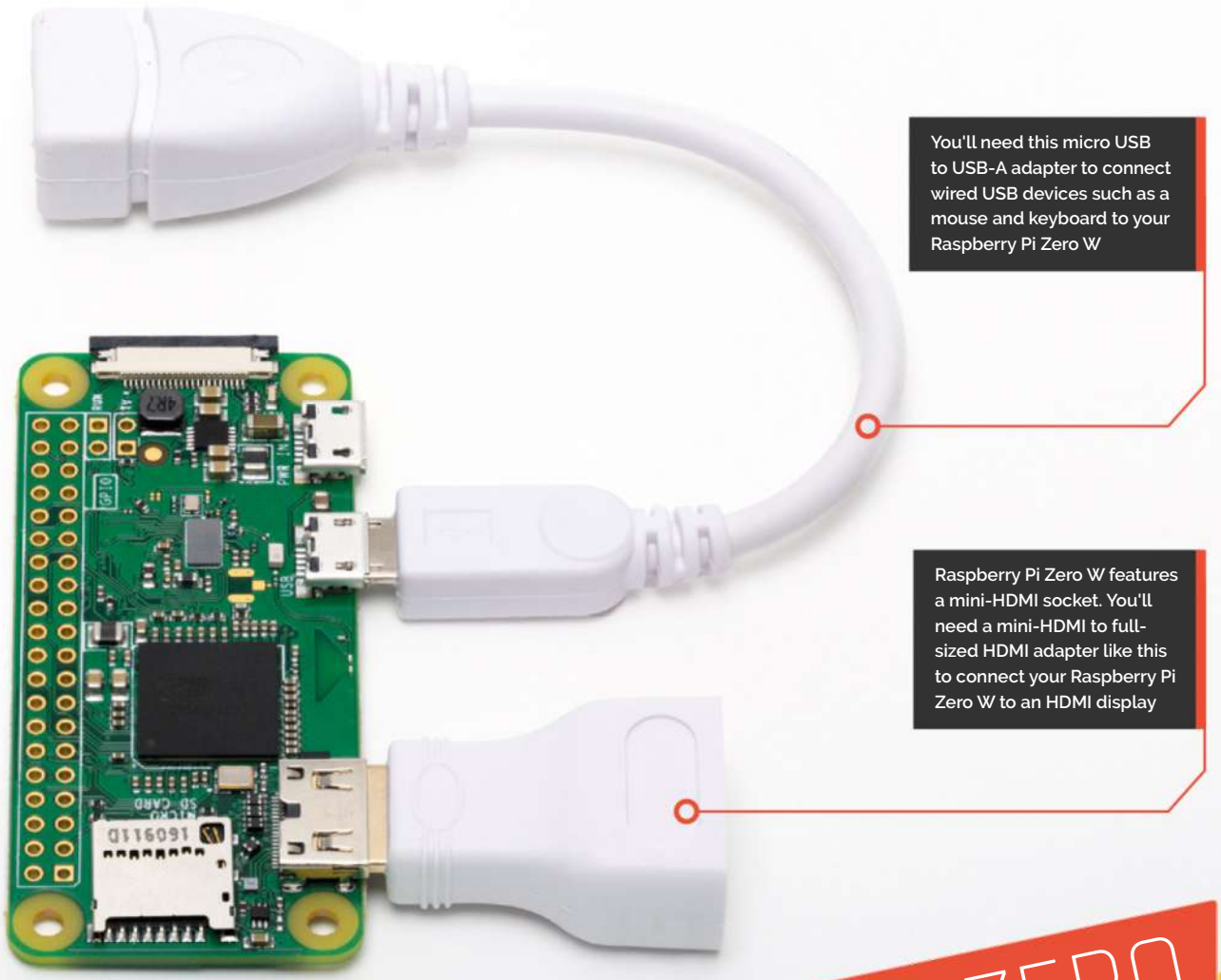
03 HDMI cable

Next, connect Raspberry Pi to your display using an HDMI cable. This will connect to one of the micro-HDMI sockets on the side of a Raspberry Pi 4, or full-size HDMI socket on a Raspberry Pi 3/3B+. Connect the other end of the HDMI cable to an HDMI monitor or television.

An HDMI cable, such as ones used by most modern televisions, is used to connect Raspberry Pi to a TV or display. You'll need a micro-HDMI to HDMI cable (or two) to set up a Raspberry Pi 4. Or a regular HDMI cable for Raspberry Pi 3B+ / 3 (or older) models



A regular wired mouse is connected to any of the USB-A sockets. A wired keyboard is connected to another of the USB-A sockets. If you have a Raspberry Pi 4, it's best to keep the faster (blue) USB 3.0 sockets free for flash drives or other components



You'll need this micro USB to USB-A adapter to connect wired USB devices such as a mouse and keyboard to your Raspberry Pi Zero W

Raspberry Pi Zero W features a mini-HDMI socket. You'll need a mini-HDMI to full-sized HDMI adapter like this to connect your Raspberry Pi Zero W to an HDMI display

SET UP RASPBERRY PI ZERO

You'll need a couple of adapters to set up a Raspberry Pi Zero computer

01 Get it connected

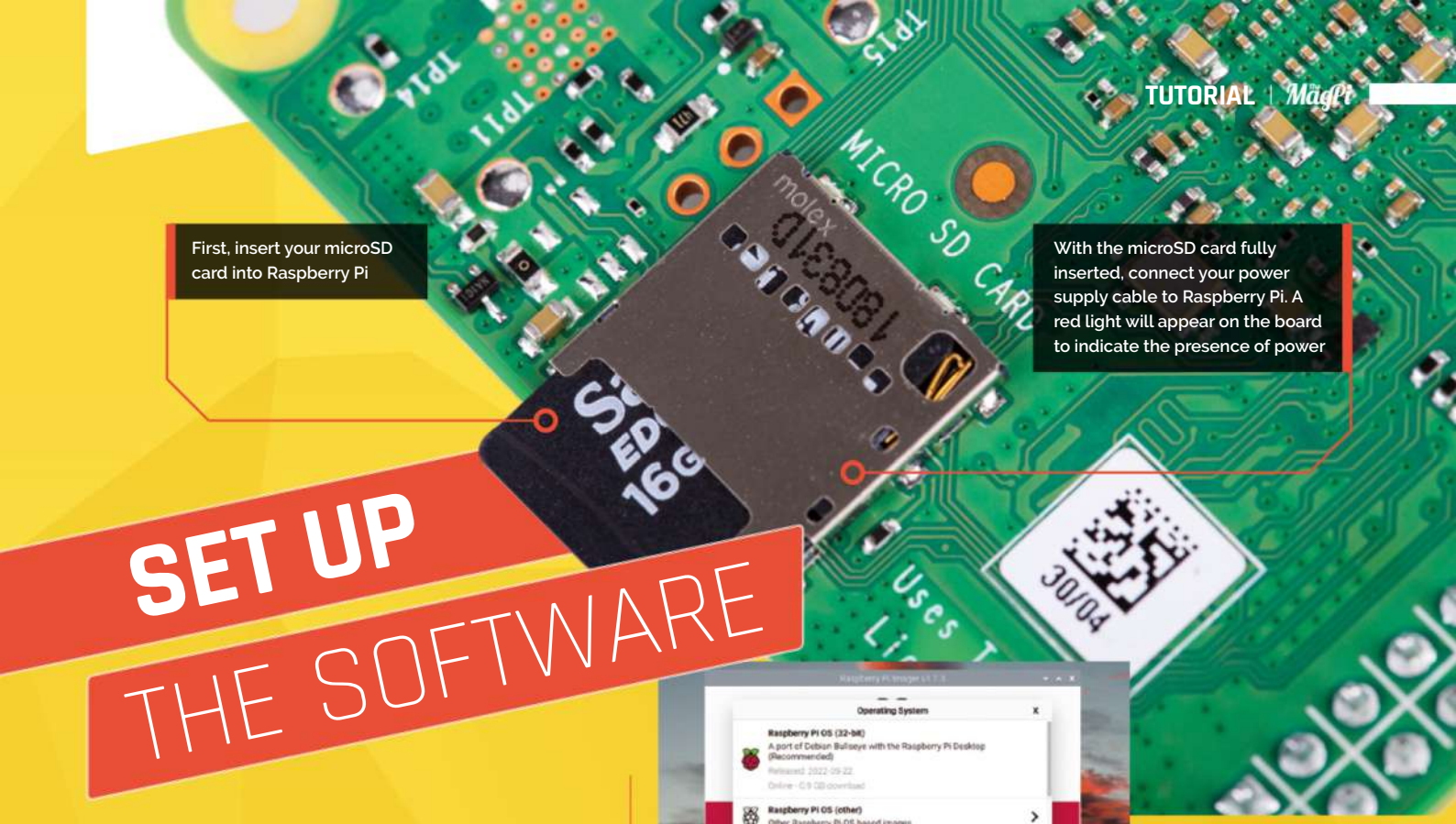
If you're setting up a smaller Raspberry Pi Zero, you'll need to use a micro USB to USB-A adapter cable to connect the keyboard to the smaller connection on the board. Raspberry Pi Zero models only have a single micro USB port for connecting devices, which means you'll need to either get a small USB hub or use an all-in-one mouse and keyboard.

02 Mouse and keyboard

You can either connect your mouse to a USB socket on your keyboard (if one is available), then connect the keyboard to the micro USB socket (via the micro USB to USB-A adapter). Or, you can attach a USB hub to the micro USB to USB-A adapter.

03 More connections

Now connect your full-sized HDMI cable to the mini-HDMI to HDMI adapter, and plug the adapter into the mini-HDMI port in the middle of your Raspberry Pi Zero. Connect the other end of the HDMI cable to an HDMI monitor or television.


First, insert your microSD card into Raspberry Pi
With the microSD card fully inserted, connect your power supply cable to Raspberry Pi. A red light will appear on the board to indicate the presence of power

SET UP THE SOFTWARE

Use Imager to install Raspberry Pi OS on your microSD card and start your Raspberry Pi

Now you've got all the pieces together, it's time to install an operating system on your Raspberry Pi so you can start using it. Raspberry Pi OS is the official software for Raspberry Pi, and the easiest way to set it up on your Raspberry Pi is to use Raspberry Pi Imager. See the 'You'll Need' box and get your kit together.

Raspberry Pi
Raspberry Pi OS
Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.
Install Raspberry Pi OS using Raspberry Pi Imager
Raspberry Pi

01 Download Imager

Raspberry Pi Imager is available for Windows, Mac, and Linux computers. You can also install it on Raspberry Pi computers, to make more microSD cards once you are up-and-running. Open a web browser on your computer and visit **magpi.cc/imager**. Once installed, open Imager and plug your microSD card into your computer.

Operating System
Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
Other general-purpose OS
Other general-purpose operating systems
Media player OS

02 Choose OS

Click on 'Choose OS' in Raspberry Pi Imager and select Raspberry Pi OS (32-bit). Click 'Choose SD card' and select the microSD card you just inserted (it should say 8GB or the size of the card next to it). Click on 'Write'. Your computer will take a few minutes to download the Raspberry Pi OS files, copy them to the microSD card, and verify that the data has been copied correctly.

Welcome to Raspberry Pi
Welcome to the Raspberry Pi Desktop!
Before you start using it, there are a few things to set up.
Press 'Next' to get started.
Cancel
Next

03 Set up Raspberry Pi

When Raspberry Pi Imager has finished verifying the software, you will get a notification window. Remove the microSD card and put it in your Raspberry Pi. Plug in your Raspberry Pi power supply and, after a few seconds, a blue screen will appear with 'Resizing Filesystem'. It will quickly vanish and be replaced by 'Welcome to Raspberry Pi'. Click on Next and follow the on-screen instructions to set up Raspberry Pi OS and start using your new computer.

Top Tip

Is your card ready?

You don't need to do this if your Raspberry Pi came with a card pre-written with Raspberry Pi OS.

You'll Need

- A Windows/Linux PC or Apple Mac computer
- A microSD card (8GB or larger)
- A microSD to USB adapter (or a microSD to SD adapter and SD card slot on your computer)
- Raspberry Pi Imager **magpi.cc/imager**

This BOOTSEL button is used to connect Pico to a computer. Hold it down when powering on and Pico appears as a drive on the other machine
A USB to micro USB cable is used to connect Pico to another computer which is used to program, debug and transfer files

SET UP RASPBERRY PI PICO

How to program your Pico microcontroller board

You'll Need

- Computer (Raspberry Pi, Mac, or PC)
- Raspberry Pi Pico or Pico W **magpi.cc/pico**
- Micro-USB cable

01 Drag and drop Pico

Raspberry Pi Pico is a miniature marvel, capable of running code and connecting to electronic devices via its GPIO (general-purpose input/output) pins. You program Pico by connecting it to a computer via USB, then transferring code files to it (or coding and debugging directly using an IDE – integrated development environment).

02 Switch to MicroPython

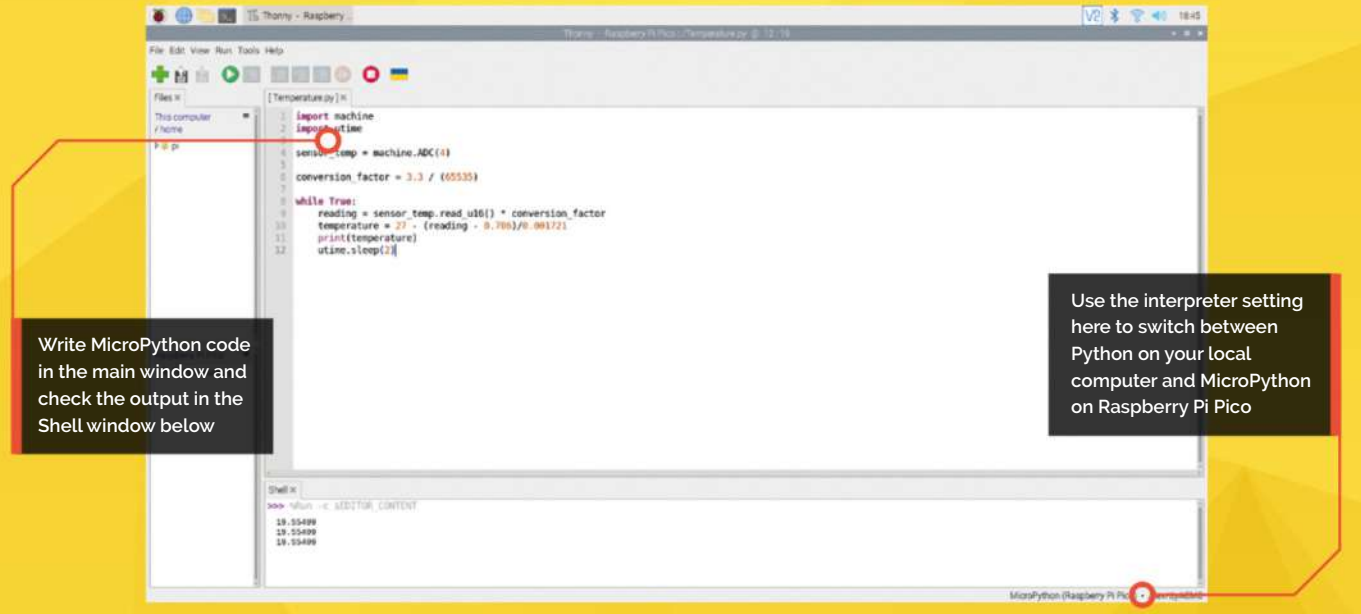
Raspberry Pi Pico is set up, by default, for use with the C/C++ Software Development Kit (SDK). The C/C++ SDK is an extremely flexible and powerful way to interact with your Raspberry Pi Pico. However, there's a more beginner-friendly method: MicroPython.

Hold down the small button on your Raspberry Pi Pico marked 'BOOTSEL' and connect Pico to your computer using the USB cable. (We are using a Raspberry Pi). Wait a few seconds, then let go of the BOOTSEL button. You will see your computer mount a removable drive. Click OK in the 'Removable medium is inserted' window to open Raspberry Pi Pico's on-board storage.

03 Install the UF2 firmware

Double-click the **INDEX.HTM** file shown in Pico's mounted storage. Your browser will open and display the Raspberry Pi Documentation web page. Select 'MicroPython', then click the option to download the correct MicroPython firmware for your board (Pico or Pico W). It's a small file, so it'll only take a few seconds.

Open File Manager and locate the.uf2 file in the **Downloads** folder. Drag-and-drop the UF2 file to the Raspberry Pi Pico's removable drive (named 'RPI-RP2'). After a few seconds, the drive will disappear as the new MicroPython firmware is recognised and installed.


Write MicroPython code in the main window and check the output in the Shell window below
Use the interpreter setting here to switch between Python on your local computer and MicroPython on Raspberry Pi Pico

CODE PICO WITH THONNY

Program Pico using a development environment

01 Open Thonny

The best way to program in MicroPython on your Raspberry Pi Pico is with the Thonny Python IDE (integrated development environment). Open the Raspberry Pi menu and choose: Programming > Thonny Python IDE.

If you are using a non-Raspberry Pi computer you may have to install Thonny. Follow the instructions at **thonny.org**.

> Thonny is normally used to write programs that run on the same computer you're using Thonny on

02 Choose interpreter

Thonny is normally used to write programs that run on the same computer you're using Thonny on; to switch to writing programs on your Raspberry Pi Pico, you'll need to choose a new Python interpreter. Look at the bottom-right of the Thonny window for the word 'Python' followed by a version number: that's your current interpreter.

Click 'Python' and look through the list that appears for 'MicroPython (Raspberry Pi Pico)' – or, if you're running an older version of Thonny, 'MicroPython (generic)'.

03 Hello World

Writing a program for your Raspberry Pi Pico is a lot like writing a program for your Raspberry Pi. You can type commands in the Shell area at the bottom of the window to have them immediately executed, or you can write a program in the main part of the window to run on-demand.

Click in the Shell area, next to the >>>> symbols, and type:

```
print("Hello, World!")
```

When you press **ENTER** at the end of the line, you'll see your Raspberry Pi Pico respond.

You are now ready to start writing your MicroPython code in the main window. Click Run to start the program on Raspberry Pi Pico. Take a read of our book, *Get Started with MicroPython on Raspberry Pi Pico* if you want to learn more about Pico programming (**magpi.cc/picobook**).

Top Tip

Update Thonny

If you don't see 'MicroPython (Raspberry Pi Pico)' in the interpreter list, you'll need to update Thonny. Open a Terminal window and type:

```
sudo apt update
&& sudo apt
full-upgrade -y
```

RASPBERRY PI SYSTEM GUIDE

ONCE YOU'VE GOT YOUR RASPBERRY PI CONNECTED UP; IT'S TIME TO START USING IT

BY **PHIL KING**

No matter which model of Raspberry Pi computer you have, you have a powerful piece of kit in your hands that can be used to learn coding, control electronic circuits (using the GPIO header), and for all sorts of projects for your home. We'll show you how to get started.

RASPBERRY PI OS

Navigate the official operating system and its applications

Click on the Raspberry Pi icon at the top left of the desktop to open the applications menu; it has several categories

When you first power up Raspberry Pi, you will be greeted by a Welcome Wizard that takes you through the initial setup process for the operating system, including locale settings, creating a user, and connecting to Wi-Fi. Once setup is complete, you are prompted to restart Raspberry Pi. After a short delay, you will then see the Raspberry Pi OS desktop.

Around the desktop

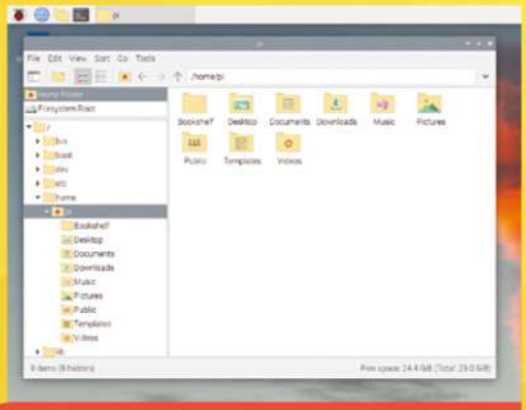
The desktop GUI for Raspberry Pi OS looks similar to that on many other computer systems. At the top is a taskbar that shows a few pinned applications on the left: by default, the Chromium web browser, File Manager, and Terminal. The taskbar will also show tabs for any currently open applications or windows.

At the right end of the taskbar are some icons for Bluetooth, Wi-Fi, and audio. Left-clicking these brings up some basic options, such as a volume slider for audio. Right-clicking the audio icon also allows you to choose the audio output (if your Raspberry Pi has a 3.5mm AV jack, or has an add-on sound card connected). Hovering over the icons will also bring up information on their status, including your Raspberry Pi's IP address for Wi-Fi.

At the far right of the taskbar is the current time (sourced from the internet). Hovering over it shows the date. Clicking it brings up a calendar.

Applications menu

At the top left of the taskbar is a Raspberry Pi icon. Clicking this brings up the applications menu. If you have installed the standard 32-bit version of Raspberry Pi OS, by default this will



The File Manager makes it easy to find folders and files in Raspberry Pi OS

SYSTEM SETTINGS

Found in the Preferences category, the Raspberry Pi Configuration tool gives you access to numerous system settings. For instance, in the Interfaces tab, you can turn various communications protocols on/off, such as VNC or SSH for connecting from a remote computer.

show the following categories: Programming, Internet, Sound & Video, Graphics, Accessories, Help, and Preferences.

If you installed the 'Full' (not default) version of Raspberry Pi OS, your applications menu will include extra software packages such as Claws Mail and LibreOffice, some of them in new categories. You can install such software manually, by going to Preferences > Recommended Software. To install an application, tick the check box to the right of it and then click Apply. Once installed, it will appear in the applications menu. Similarly, to remove a previously installed application, untick it from the list.

Even more software packages are available by going to Preferences > Add/Remove Software. As with the Recommended Software tool, you can view applications by category and tick the box next to a package to install it.

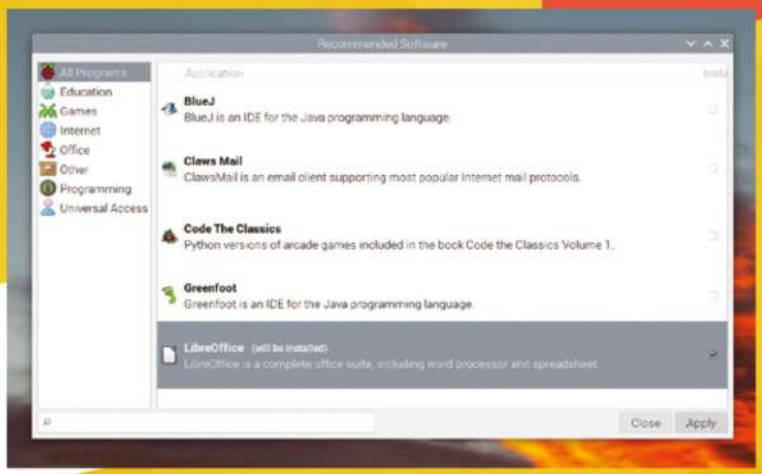
You can also install extra software in the Terminal application, using the APT package manager. The Terminal is a powerful command-line interface for entering all sorts of commands – see this issue's tutorial for more details.

File and web browsing

The File Manager on Raspberry Pi OS enables you to browse files and folders stored on the microSD card, along with removable storage media connected via USB. There are two main view modes: 'icons' and 'list'. You'll find the most useful folders (also known as 'directories') in your home folder, including **Documents**, **Downloads**, and **Pictures**. The **Bookshelf** folder enables you to download and view your favourite Raspberry Pi Press magazines, including *The MagPi*.

> You can also install extra software in the Terminal

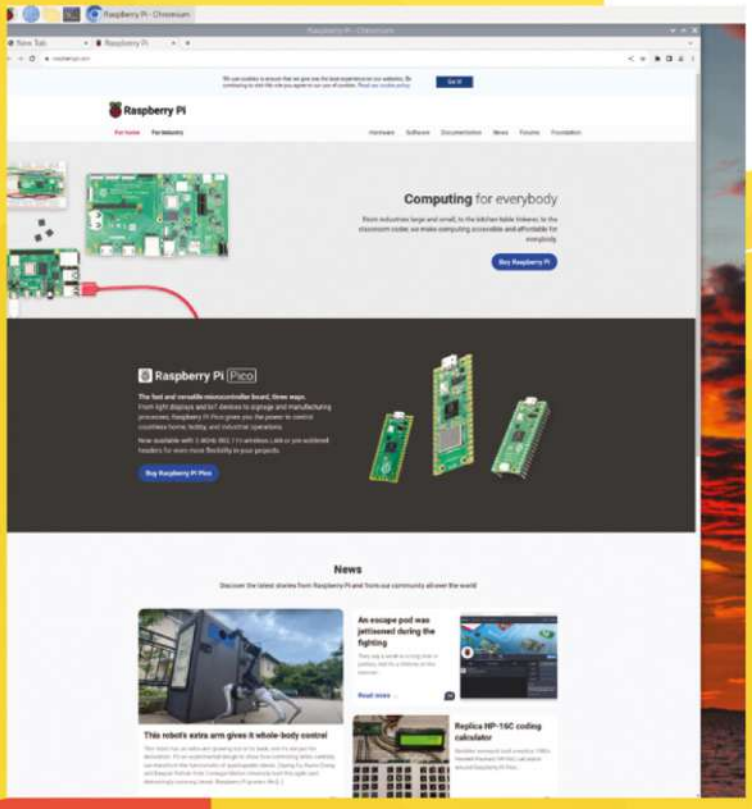
For surfing the web, Raspberry Pi OS comes with the Chromium Web Browser pre-installed. This is the open-source project behind the Google Chrome browser, so will look fairly familiar to many users. At the top is an omnibox to type URLs or search terms (using the DuckDuckGo search engine by default). You can visit any site you like, including YouTube to watch videos – alternatively, you can view and listen to media in VLC Media Player, including web streams.



Installing new applications with the Recommended Software tool

ICON SIZE

Are the icons and menus on the desktop too small? This can happen when you connect Raspberry Pi to a large hi-res screen, such as a 4K TV. To fix it, right-click the desktop and choose Desktop Preferences. In the Defaults tab, click the Set Defaults button next to 'For large screens'.



▲ Surf the web with the Chromium web browser in Raspberry Pi OS

> Notifications appear in the taskbar when new updates are available

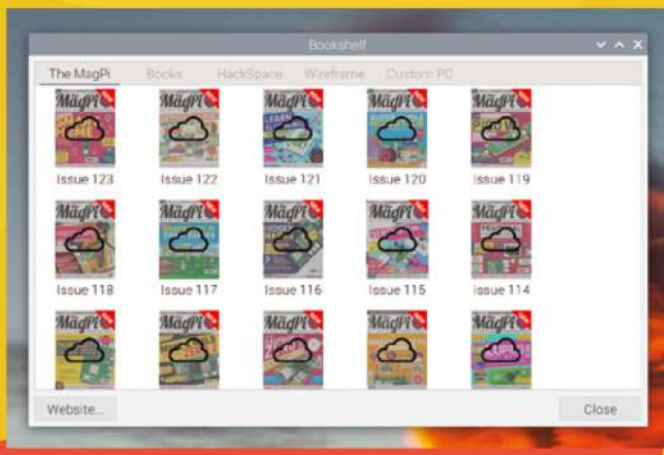
System update

For access to the latest features and important security measures, you will want to keep Raspberry Pi OS up to date. Fortunately, this is made easy using the update notifications that appear in the taskbar whenever new updates – for the system or applications – are available. You can then opt to install them from the dialog window that appears.

WIRELESS CONNECTION

If you haven't already set your Wi-Fi connection in the Welcome Wizard when you first powered up Raspberry Pi, you can do so by clicking the Wi-Fi symbol in the top menu bar of the desktop, choosing your network from the list, and entering its password.

If you want to reboot or shut down your Raspberry Pi, select Shutdown from the applications menu, then select the desired option. To avoid the possibility of losing data, it's advisable to shut down your Raspberry Pi before disconnecting the power.



▲ From the Bookshelf, you can download and read issues of *The MagPi* magazine, Raspberry Pi Press books, and other magazines

DUAL SCREEN SETUP

If you are using a Raspberry Pi 4 or 400, it's possible to connect it to two monitors. When you power up, only one screen (the primary display) will have the taskbar at the top. You can switch the primary display and alter the screen arrangement in Preferences > Screen Configuration, where options for resolution and frequency are also available.

PYTHON PROGRAMMING

Start learning to code using the popular Python programming language

Your Raspberry Pi is ideal for learning how to code. We'll take you through the basics of Python programming. In the desktop applications menu, go to Programming > Thonny Python IDE.

Thonny starts up in simple mode by default. Along the top are icons including Load, Save, Run, and Stop. The main panel is where you will write lines of Python code. At the bottom is the Shell panel where you'll see the output from your programs, and any error messages resulting from a running program.

Thonny - /home/pi/Documents/Print2.py
issue = 125
magazine = "The MagPi"
print("This is issue", issue, "of", magazine)
This is issue 125 of The MagPi

Using variables with text strings to print a phrase in the Thonny IDE

01 Hello world

Let's start with the simplest of programs: printing a message to the Shell. In the main Thonny panel, add the following line:

```
print("Hello world!")
```

When using a print statement in Python 3, brackets must be used to enclose the output (as with any other function). The double quotes are used to enclose a text message (known as a string). Run the program (you'll be prompted to save it) and it will print the message to the Shell. Try changing the text between the double quotes and rerunning it.

02 Variables

Variables are used to store values that may change in your program. You can define one and set it to a value using the = symbol, such as:

```
issue = 125
```

Here, we have set the variable called 'issue' to 125. We can also set a variable to store a text string:

```
magazine = "The MagPi"
```

We can then add these variables to a `print` statement, along with other text strings, separated by commas:

```
print("This is issue", issue, "of",
magazine)
```

03 Maths operators

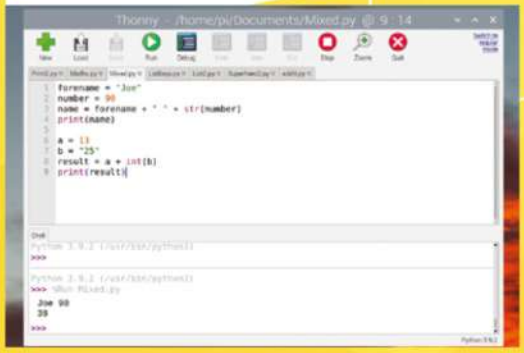
You can use maths operators to add (+), subtract (–), divide (/), and multiply (*) numbers. For instance:

```
a = 7
b = 12
c = 39
print(a+b*c)
```

Note that the BODMAS (aka PEMDAS) rule applies in Python, so `b` and `c` are multiplied

LEARN PYTHON

Take a look at *Create Graphical User Interfaces with Python*. This 156-page book is suitable for everyone, from beginners to experienced Python programmers who want to explore graphical user interfaces (GUIs).
magpi.cc/pythongui



You can use convert numbers to strings and vice versa

together before adding `a`. You can also add text strings together:

```
forename = "Joe"
surname = "Bloggs"
name = forename + " " + surname
print(name)
```

Note that we added a space between the forename and surname in our `name` string. You can't simply add a number variable to a string, however, but you can use the `str()` method to convert a number to a string.

```
forename = "Joe"
number = 90
name = forename + " " + str(number)
print(name)
```

You can also convert a suitable string (all digits) to a number:

```
a = 13
b = "25"
result = a + int(b)
print(result)
```

04 Lists and loops

In Python, lists can be used to store variables of any type, separated by commas. These can then be accessed individually using an index number.

```
fruits = ["apple", "banana", "cherry",
"date"]
print(fruits[1])
```

NUMBER TYPES

There are two main types of number in Python: integers (whole) and floats (with a decimal point). You can convert one type (or a string) to the other with the `int()` and `float()` methods.

The index number is given in square brackets. Since Python starts counting from zero, however, the output of this print statement will be 'banana'. You can add extra items to the end of a list using the `append` method:

```
fruits = ["apple", "banana", "cherry",
"date"]
fruits.append("elderberry")
fifthfruit = (fruits[4])
print(fifthfruit)
```

Typically, a list will be used with a `for` loop, to print out a range of values.

```
fruits = ["apple", "banana", "cherry",
"date", "elderberry"]
for i in range (5):
    print(fruits[i])
```

Note that the lines of code to be looped must be indented (by four spaces). This example will print out all the fruits in the list, as the `for` loop iterates with values of 0 to 4 for `i`. Alternatively, you could set the start and end of the range.

```
for i in range (1,3):
    print(fruits[i])
```

Since the loop is not executed with the end value of 3, this only prints out 'banana' and 'cherry'. Loops are useful for a variety of purposes. An alternative to a `for` loop is to use `while`. This iterates the loop so long as the condition is met.

```
counter = 0
while counter < 4:
    print(fruits[counter])
    counter = counter + 1
```

05 Conditional statements

When your create programs, you will often want to test a condition and have your program react differently depending on whether it is true or false. For this, we use `if` and `else` statements.

```
print("What is your name?")
name = input()
if name == "Peter Parker":
    print("You are Spider-Man!")
else:
    print("You are not Spider-Man.")
```

NO-INPUT FUNCTION

Note that not all functions need an input variable, in which case it will be defined and called with empty brackets.

Here, we are using the `input` method to get text input from the user. We then use an `if` statement to check whether it matches a particular string – note that `==` is used to check whether values are equal. If it does, the following indented line(s) of code is executed. If not, the indented line(s) under the `else` statement is executed.

You can set up a series of conditions by using `elif` (short for 'else if') statements.

```
if name == "Peter Parker":
    print("You are Spider-Man!")
elif name == "Bruce Wayne":
    print("You are Batman!")
else:
    print("You must be a different superhero.")
```

06 Functions

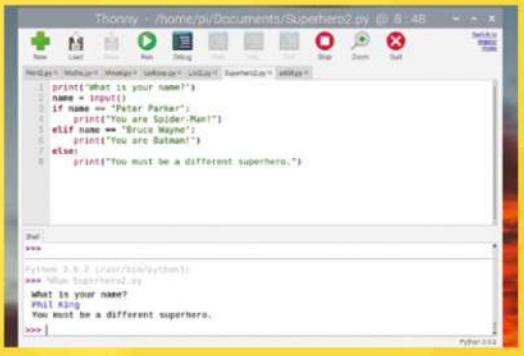
Instead of having to repeat the same block of code each time you want to do something, you can define it as a function and 'call' it any time you want to run it.

```
def addit(temp):
    f = c * 1.8 + 32
    print("In Fahrenheit, that is:", f)
```

We have defined our function with an input variable of `temp` (it doesn't matter what it's called). Now, let's call it from a `while True:` infinite loop. Note that the user input text string of `c` is turned into a floating-point number before the function is called with `addit(c)`.

```
while True:
    print("What is the temperature in Celsius?")
    c = float(input())
    addit(c)
```

Note that inputting anything other than a number will cause the program to crash.



Using conditional statements to determine what kind of superhero you are

We can use `try` and `except` clauses to check it's a number before calling the function:

```
while True:
    print("What is the temperature in Celsius?")
    try:
        c = float(input())
        addit(c)
    except ValueError:
        print("This is not a number. Please enter a valid number")
```

That concludes this introduction to coding Python. There are many free online resources and courses available to help you learn more.

LEARN PROGRAMMING

PROJECTS

The Raspberry Pi Foundation website has a bunch of great projects that will help you start writing code and get going with digital making.

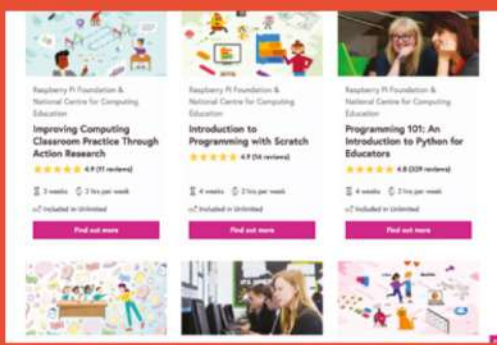
magpi.cc/projects

Learn to code with Scratch
Getting started with Raspberry Pi
Learn to code with Python
Getting started with Raspberry Pi Pico
Protect our planet
Look after yourself

FUTURE LEARN

The Raspberry Pi Foundation has teamed up with FutureLearn to provide interactive courses, most of which are in Python (and there's a great Scratch to Python course). The courses are structured to last three to four weeks and only require a few hours a week to learn.

magpi.cc/futurelearn


Improving Computing Classroom Practice Through Action Research
Introduction to Programming with Scratch
Programming 101: An Introduction to Python for Educators

ELECTRONIC CIRCUITS

Build electronic circuits and control them with Python programs

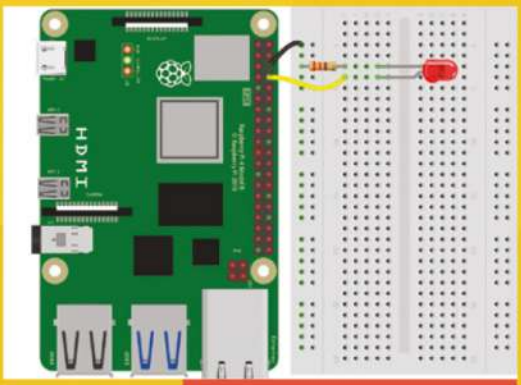
Raspberry Pi's GPIO header enables you to connect it up to electronic circuits and control them using code, also known as 'physical computing'. Components can be outputs such as LEDs and buzzers, or inputs such as push-buttons and sensors. It is best practice to build your circuits with Raspberry Pi powered off.

01 Blink an LED

The 'Hello world' of physical computing is to make an LED blink, so we'll start with that. Build your LED circuit on the breadboard as shown in **Figure 1**. The LED's legs are placed in different breadboard columns. The longer leg is the anode (positive); we connect its column to GPIO 14, so we can switch it on and off. The shorter leg is connected by a resistor to a long rail on the breadboard, which in turn is connected to a GND (ground) pin on Raspberry Pi. A resistor is essential to limit the current flowing through the LED so there is no chance it will burn out.

Now, in Thonny IDE, let's create a Python program – as in the **led.py** listing (overleaf) – to make the LED turn on and off repeatedly. To reduce the need for boilerplate code, we're using the GPIO Zero Python library, which we import at the top along with the `sleep` method from the time library. We then create an object named `led` (it could be whatever you like) as an LED set to our GPIO 14 pin.

Finally, we use a `while True:` infinite loop to turn the LED on and off, with a one-second `sleep` delay between each – try altering the delay values to slow down or speed up the LED blinking.



▲ **Figure 1:** The wiring diagram for connecting an LED to Raspberry Pi

YOU'LL NEED

- Breadboard
- LED (any colour)
- Push-button
- Active piezo buzzer
- 2 × 330Ω resistors
- Jumper wires (male-male and male-female)

02 Push to light

Next, we'll add an input device to our circuit in the form of a push-button switch. We'll then create a program so that the LED only lights up when the button is pressed.

Connect the push-button to the circuit, straddling the dip in the middle of the breadboard, as shown in **Figure 2**. On one side, we connect one leg to the ground rail; the other leg via a resistor to GPIO 23. This means that the pin will be high (3.3V) when the button is pressed, but low (0V) otherwise.

Enter the code from the **led_button.py** listing. Note that we import an extra method, `Button`, from GPIO Zero. We then create an object (named `button` here) as a `Button` for our GPIO 23 pin.

There are several ways of reading a button with GPIO Zero; here, we're using the `is_pressed` method. In a `while True:` infinite loop, we use an `if` conditional statement to check whether the button has been pressed, in which case we light the LED; if not, the LED is turned off by line of code under the `else` statement.

03 Add sound

Another type of output device you can connect is a piezo buzzer, which can make beeping noises. Note that there are two types of piezo buzzer: active and passive. Here we're using the active type – the circuit and code would need altering for a passive one. As in **Figure 3**, the

PROJECT RESOURCES

For more ideas for simple electronics projects using Raspberry Pi, visit the Raspberry Pi Foundation site: **magpi.cc/projects**. Click 'Browse all projects', then select Raspberry Pi from the 'Any hardware' dropdown.

longer, positive leg is connected to GPIO 12. The shorter, negative leg is connected to the ground rail of the breadboard.

Enter the code from the **buzzer.py** listing. In the code, we import the `Buzzer` method from GPIO Zero. We then create an object (named `buzz` here) as a `Buzzer` on GPIO 12. Just as we did when

> We turn the buzzer on and off to make it beep

blinking the LED, we turn the buzzer on and off to make it beep – this time in a `for` loop that iterates five times.

04 Putting it together

In the **doorbell.py** code listing, we create a primitive doorbell: when the button is pressed, the LED blinks and the buzzer beeps five times. This time we create a `ring` function for the doorbell ringing and flashing. We then assign GPIO Zero's `when_pressed` method to

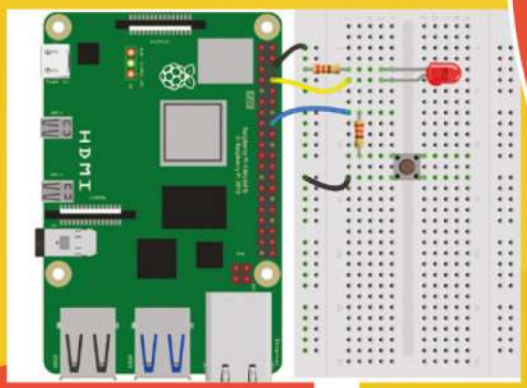


Figure 2: We have added a push-button switch to our circuit

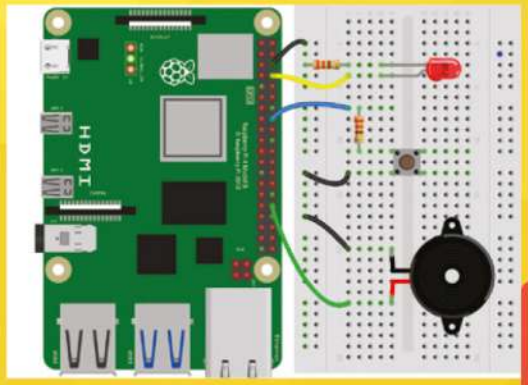
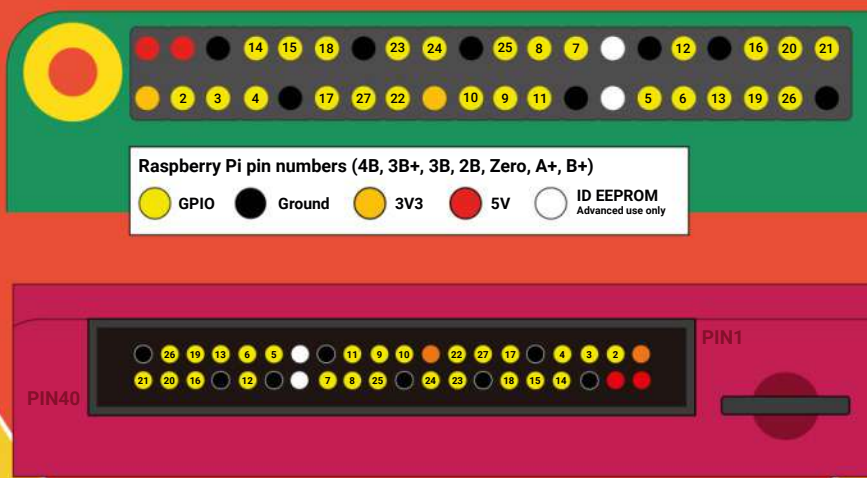
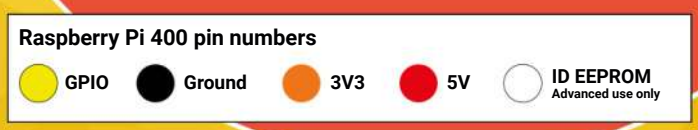


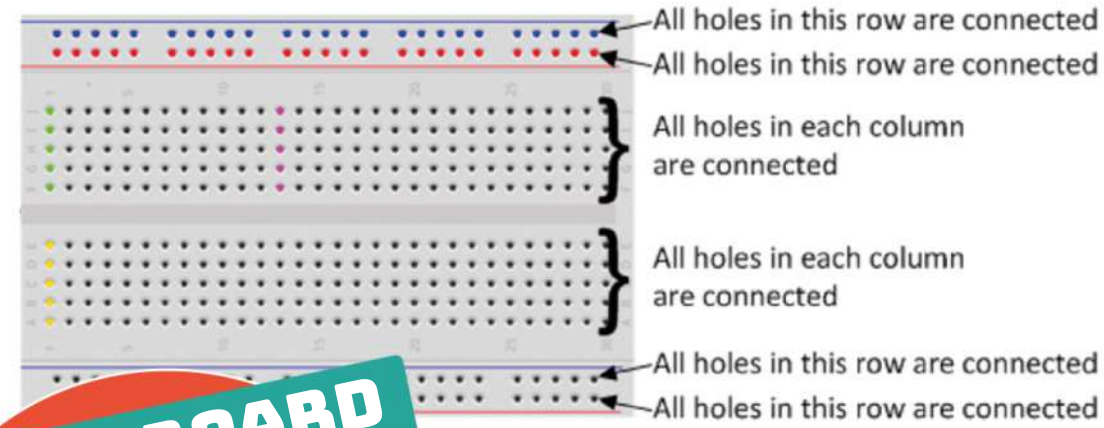
Figure 3: The complete circuit, with an active piezo buzzer added

GPIO PINS

The 40-pin GPIO (general-purpose input/output) header enables a Raspberry Pi computer to be connected to electronic circuits (as well as HATs and other add-on boards). There are 26 numbered GPIO pins that can be used to switch a connected component on/off, or to read a digital input. Other pins include 3V3 (3.3V) and 5V power pins and several GND pins for a ground (earth) connection. While the layout is the same on all modern Raspberry Pi models, note that on a Pi 400 the header is turned upside down.






All holes in this row are connected
All holes in this row are connected
All holes in each column are connected
All holes in each column are connected
All holes in this row are connected
All holes in this row are connected

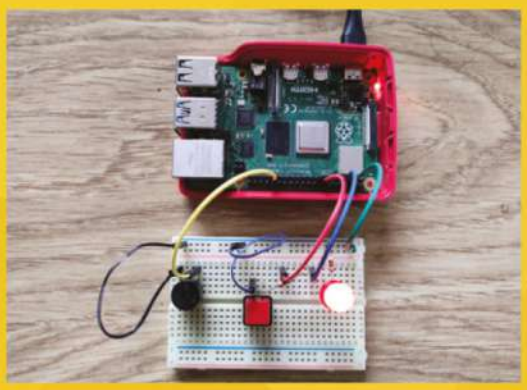
BREADBOARD CONNECTIONS

A breadboard makes it easy to make solderless connections between electronic components. There are different sizes available, but they all work in a similar way. The five holes in each column are connected together. Most breadboards also have two long 'rails' on each side – the holes in each rail are connected, so they're often used to connect multiple components to one power or ground pin.

PULL UP OR DOWN

In our button circuit example, the pin is pulled high (3.3 V, logic 1) when the button isn't pressed, which is the default setting for the `Button` method in GPIO Zero). Pressing the button connects the pin to ground and pulls it down to 0 V (logic 0).

Alternatively, you could set the pin to be pulled down by default, with `button = Button (23, pull_up=False)`. In this case you would wire the other leg of the button to 3V3 instead of GND, to pull GPIO 23 up to high when it's pressed.



▲ Once the button is pressed, the function is called to blink the LED and sound the buzzer

call the function when the button is pressed. To stop the program then exiting, we use the `pause` method from the signal library.

If you want to extend this project, you could try using GPIO Zero's `TonalBuzzer` method to play a rudimentary (and buzzy) tune on the buzzer! For more details, see the complete GPIO Zero documentation at **magpi.cc/gpiozero**.

DOWNLOAD THE FULL CODE: magpi.cc/github

led.py

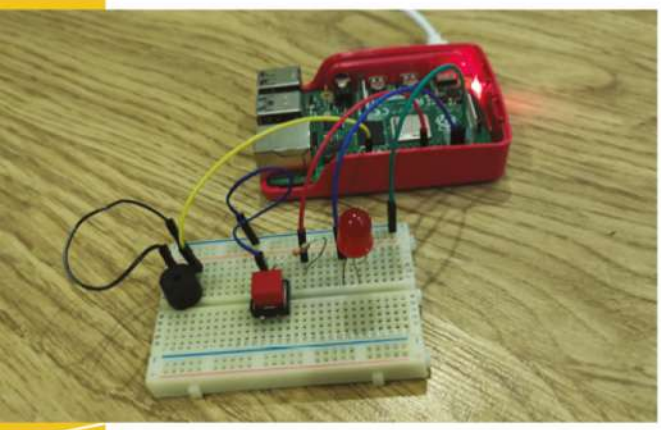
Language: **Python 3**

```
001.  from gpiozero import LED
002.  from time import sleep
003.
004.  led = LED(14)
005.
006.  print("Blinking the LED")
007.
008.  while True:
009.      led.on()
010.      sleep(1)
011.      led.off()
012.      sleep(1)
```

led_button.py

Language: **Python 3**

```
001.  from gpiozero import LED, Button
002.
003.  led = LED(14)
004.  button = Button(23)
005.
006.  while True:
007.      if button.is_pressed:
008.          led.on()
009.      else:
010.          led.off()
```



▲ The circuit for our basic doorbell, with components on a breadboard connected to Raspberry Pi's GPIO header

doorbell.py

Language: **Python 3**

```
001.  from gpiozero import LED, Button, Buzzer
002.  from time import sleep
003.  from signal import pause
004.
005.  led = LED(14)
006.  button = Button(23)
007.  buzz = Buzzer(12)
008.
009.  def ring():
010.      for i in range(5):
011.          buzz.on()
012.          led.on()
008.          sleep(0.5)
009.          buzz.off()
010.          led.off()
011.          sleep(0.5)
012.
008.  button.when_pressed = ring
009.
010.  pause()
```

buzzer.py

Language: **Python 3**

```
001.  from gpiozero import Buzzer
002.  from time import sleep
003.
004.  buzz = Buzzer(12)
005.
006.  print("Beeping the buzzer")
007.
008.  for i in range(5):
009.      buzz.on()
010.      sleep(0.5)
011.      buzz.off()
012.      sleep(0.5)
```

RASPBERRY PI PROJECTS

There are so many different things you can do with a Raspberry Pi

With its ability to run an operating system, fast processor, wireless connectivity, and a GPIO header for connecting electronics, your Raspberry Pi can be used in a vast variety of projects. To help inspire you, we've collected just a few of the more popular uses here, but there are so many more.

With LibreELEC/Kodi, you can stream media and even watch live TV with a Raspberry Pi TV HAT attached

MEDIA CENTRE

One of the most popular uses for a Raspberry Pi is to use it to play video and audio on your TV. There's even a dedicated media player OS for this, LibreELEC (**libreelec.tv**), which incorporates the Kodi media player (**kodi.tv**). It's available under the 'Media Player OS' category in Raspberry Pi Imager. Alternatively, you can install Kodi manually within Raspberry Pi OS.

Either way, you will be able to access files on any network-attached storage you have, as well as popular streaming services including YouTube, Netflix, BBC iPlayer, and Spotify – via official or unofficial add-ons. An optional mini remote control makes navigating the menus easier. With the addition of a Raspberry Pi TV HAT, you can even watch live digital TV and record it!

For more details, check out our Build the Ultimate Media Centre feature in issue 102 (**magpi.cc/102**).

RETRO GAMING

Another classic use for Raspberry Pi is to turn it into a retro games machine. The easiest way to start playing retro games is to install the specialist RetroPie OS (**retropie.org.uk**), available under the 'Emulation and Game OS' category in Raspberry Pi Imager. RetroPie includes emulation software for over 50 classic consoles and computers, including NES, SNES, N64, Mega Drive/Genesis, PlayStation 1 and 2, and MAME for classic coin-ops.

While you can simply run RetroPie on any TV or monitor and hook up a USB joystick, an alternative is to build your own arcade machine based around it. Kits are also available, including Pimoroni's Picade (**magpi.cc/picade**).

For more info on the topic, check out our book, *Retro Gaming With Raspberry Pi* (**magpi.cc/retrogaming**).

Raspberry Pi is ideal for retro gaming. You can even build a mini arcade machine such as the Picade

The Raspberry Pi High Quality Camera can be used with different lenses

A wide range of robotics kits are available, including Pimoroni's Trilobot

CAMERA

While you can connect a standard USB webcam to Raspberry Pi, the official Camera Module or High Quality Camera offer a lot more control over your image capturing. Both types connect via Raspberry Pi's CSI (camera serial interface) port.

The ultra-compact Camera Module V2 (**magpi.cc/camera**) comes in two variants: standard and NoIR – the latter lacks an IR filter, so can be used to capture images at night (with infrared LED illumination). The High Quality Camera has a larger 12.3MP sensor and can be fitted with any C- or CS-mount camera lens.

There are many uses for either camera type, from setting up a wildlife camera trap to a multi-camera CCTV system. For more ideas, see our *Raspberry Pi Camera Guide* (**magpi.cc/camerabook**) and the photography feature in issue 118 (**magpi.cc/118**).

HOME AUTOMATION

With its Wi-Fi and GPIO connectivity, Raspberry Pi is ideal for use in a smart home automation system. With the addition of a relay add-on board, such as the Automation HAT (**magpi.cc/automationhat**), it can enable you to switch lights and many standard domestic appliances on and off remotely. Or you can add motors and servos to things like curtains and blinds to automate them.

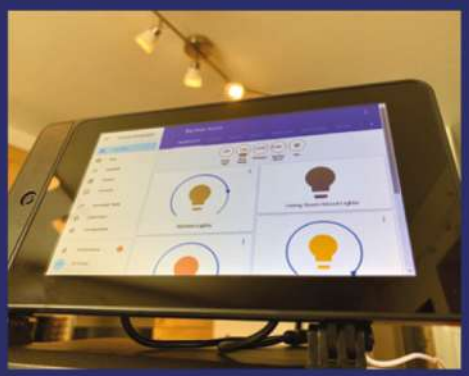
Everything can be controlled via a web dashboard such as Home Assistant (**home-assistant.io**), available as an OS image to install on Raspberry Pi. To start setting up your smart home, read the three-part tutorial series starting in issue 99 (**magpi.cc/99**).

ROBOTICS

Raspberry Pi computers have long been used in homemade robots, thanks to their compact size and processing power, enabling robots to navigate their surroundings autonomously, using sensors such as ultrasonic range-finders to detect obstacles to avoid.

A mini industry has built up around the creation of Raspberry Pi robotics kits, including a variety of wheeled vehicles, robotic arms, and even artificial pets such as the Petoi Bittle robot dog. One of the easiest kits for beginners is the CamJam EduKit #3 (**magpi.cc/edukit3**), which enables you to build a two-wheeled robot – you can even use the box as a makeshift chassis.

Alternatively, you can create your own custom robot using the wide range of motors, servos, motor driver boards, chassis, and sensors available. See our robotics starter guide in issue 120 (**magpi.cc/120**).



View and control your smart home appliances on the Home Assistant web dashboard

DESKTOP PC

Looking for an inexpensive computer to work from home? The more powerful models such as Raspberry Pi 4 and 400 are viable as a desktop PC replacement. Raspberry Pi 400 even has its own integrated keyboard and comes with a mouse, so you're all set.

Many online services for team collaboration are available, such as Google Docs and Slack. Raspberry Pi OS also enables you to install the free and open-source LibreOffice productivity suite, which includes apps for word processing, spreadsheets, presentations, databases, and more. See our Work From Home feature in issue 93 (**magpi.cc/93**).

MAGIC MIRROR

One of the most popular projects using Raspberry Pi is to make a 'magic mirror'. Built using one-way reflective glass on top of a monitor, it can display useful information such as traffic and weather as you brush your hair in the morning. Fancy building a mini magic mirror? Check out our feature in issue 122 (**magpi.cc/122**).

Get started with the command line

Learn how to use text-based commands in Raspberry Pi OS

MAKER

Phil King

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

Get Started with the Command Line book

To discover more useful commands and tips for using the command line on Raspberry Pi, check out our Conquer the Command Line book, available as a free PDF: **magpi.cc/commandline**

The standard version of Raspberry Pi OS has a user-friendly graphical user interface, but there are a few limitations to what you can do with it. That's where the command line comes in, enabling you to get under the hood of the system and unleash the power of text-based commands.

You can access the command line in several ways. The simplest is to open the Terminal app in Raspberry Pi OS (**Figure 1**) – either by clicking its icon (a black rectangle) in the taskbar or opening it from the menu – it's in the Accessories category. Alternatively, you can open a virtual console by pressing **ALT+CTRL+F1**. You'll also be taken straight to the command line if you install the Lite version of Raspberry Pi OS, without the desktop – or if you opt to 'Boot to CLI' in the Raspberry Pi Configuration tool.

Another command way to access the command line is to use SSH (Secure Shell) – see the guide at **magpi.cc/ssh** – to access your Raspberry Pi remotely from another computer, which is often very useful.

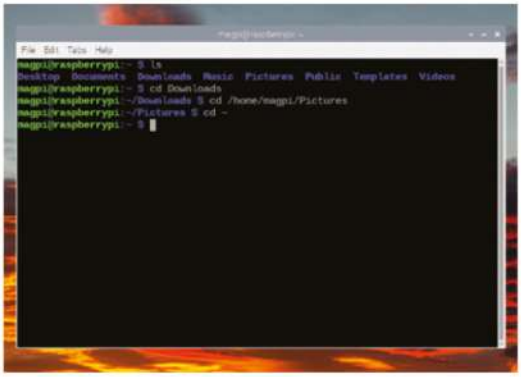


Figure 1: You can access the command line from the Terminal app, as shown here, or by switching to a virtual console or SSHing in from another computer

01 First steps

When you first access the command line, via the Terminal or other means, you will see a prompt starting with your username followed by '@' and the hostname (raspberrypi by default) with a colon. For instance:

```
magpi@raspberrypi:
```

After this, you'll see the current directory you're in, followed by a '$'. You'll start off in the user's home directory, indicated by '~'.

'Directory' is another word for folder and they correspond to the ones seen in the File Manager on the desktop. So, within your user's home directory (which is located at **/home/[username]**), you'll find the usual default directories such as **Downloads** and **Pictures**. To see what's in the current directory, we use the `ls` command:

```
ls
```

This is short for 'list' and shows the contents of the current directory.

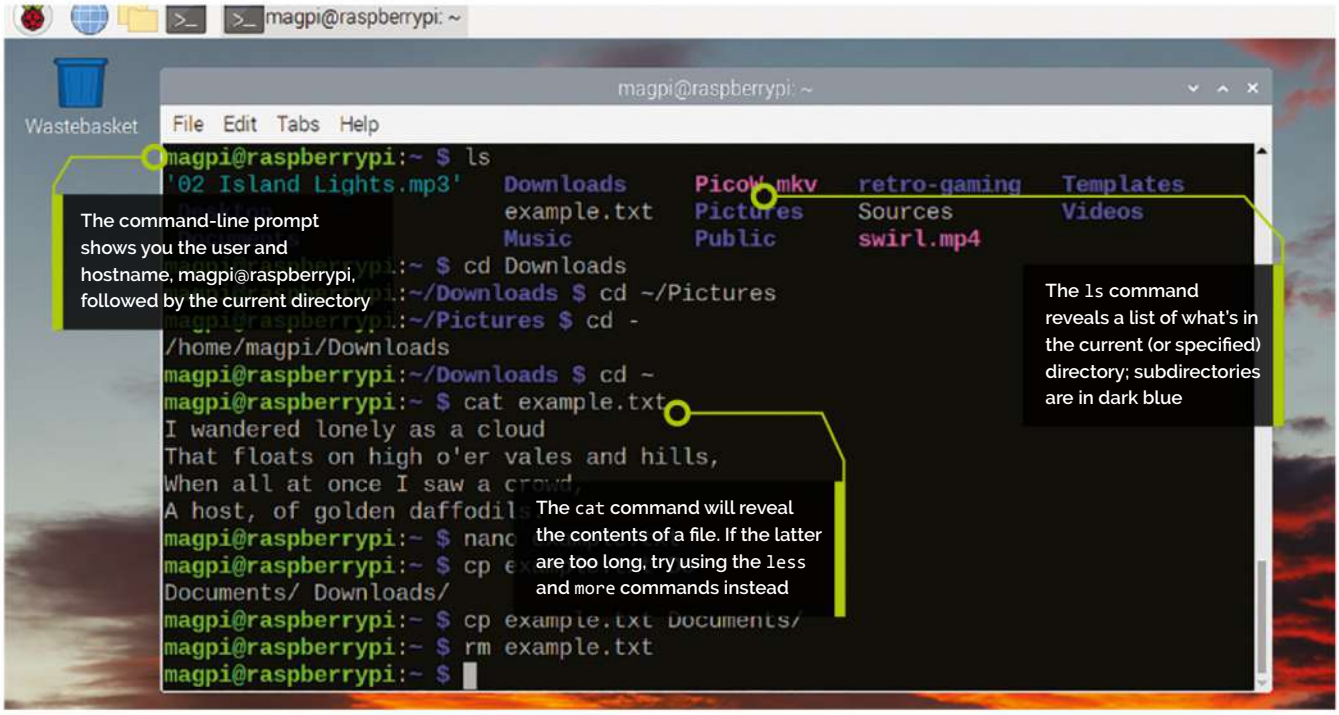
02 Navigating directories

To change to a different directory, the `cd` command is used. For instance:

```
cd Downloads
```

You will now see '~/Downloads' before the '$' in the prompt. Note that to visit another directory in your home folder from here, you'll have to enter the full path, such as:

```
cd home/magpi/Pictures
```


magpi@raspberrypi: ~
Wastebasket
File Edit Tabs Help
magpi@raspberrypi:~ $ ls
'02 Island Lights.mp3' Downloads Pico8.mkv retro-gaming Templates
example.txt Pictures Sources Videos
Music Public swirl.mp4
magpi@raspberrypi:~ $ cd Downloads
magpi@raspberrypi:~/Downloads $ cd ~/Pictures
magpi@raspberrypi:~/Pictures $ cd -
/home/magpi/Downloads
magpi@raspberrypi:~/Downloads $ cd ~
magpi@raspberrypi:~ $ cat example.txt
I wandered lonely as a cloud
That floats on high o'er vales and hills,
When all at once I saw a crowd,
A host, of golden daffodils
magpi@raspberrypi:~ $ cp example.txt Documents/
Documents/ Downloads/
magpi@raspberrypi:~ $ rm example.txt
magpi@raspberrypi:~ $
The command-line prompt shows you the user and hostname, magpi@raspberrypi, followed by the current directory
The ls command reveals a list of what's in the current (or specified) directory; subdirectories are in dark blue
The cat command will reveal the contents of a file. If the latter are too long, try using the less and more commands instead

Or you can use the shorthand of '~' in place of 'home/magpi':

```
cd ~/Pictures
```

In addition, `cd ..` takes you up one directory level (to the parent directory) while `cd -` returns you to the previous directory.

If you forget what directory you are in, you can find it with:

```
pwd
```

This stands for 'print working directory'.

03 File manipulation

Now we can find our way around the file system, let's create a new text file using the `touch` command. In the home directory, enter:

```
touch example.txt
```

Enter the `ls` command and it will show up in the list. To edit it, we can use the nano text editor:

```
nano example.txt
```

Enter a few lines of text (**Figure 2**) and then close it with **CTRL+X** followed by **Y** and **ENTER** to save the changes. We can see what's in a text-based file using the `cat` command.

```
cat example.txt
```

If we want to move or copy the file to another directory, we use the `mv` or `cp` command respectively, followed by the path of the target directory. For instance:

```
cp example.txt ~/Documents
```

If you `cd` to the **Documents** directory and `ls`, you will see the copy of the file there. You can delete a file using the `rm` command:

```
rm example.txt
```

To search for a file, you can use the `find` command with a selected directory, `-name` option, and filename (which can include the * wildcard for 'any characters'):

```
find /home/magpi -name "e*.txt"
```

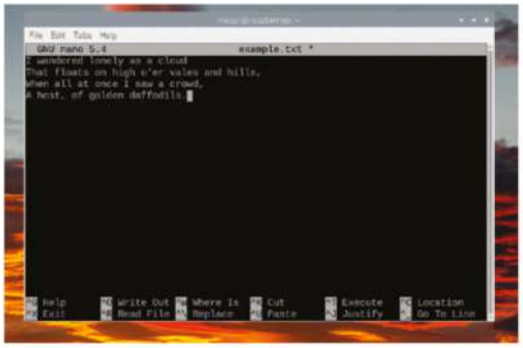


Figure 2: The nano text editor enables you edit text-based files, including programs that you can run

Top Tip

Auto-complete

The **TAB** key is your friend on the command line. Start typing a command, file, or directory name and then press **TAB** to either auto-complete it or list the possibilities. You can also use the up and down arrows to explore your command history and quickly re-enter previous commands.

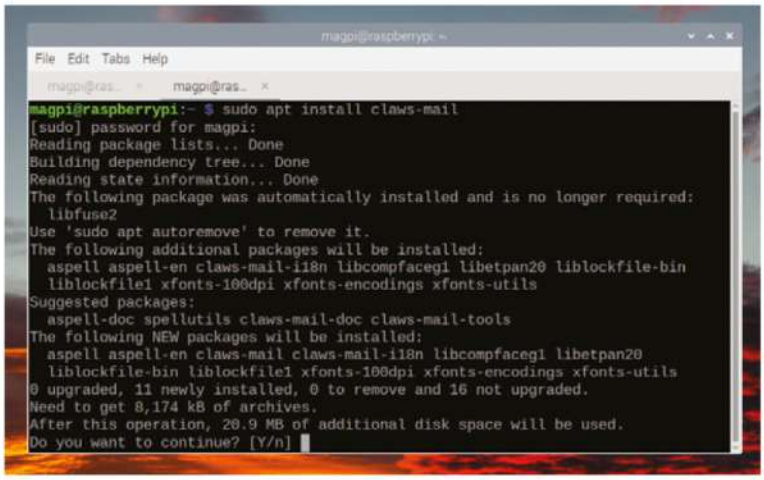

File Edit Tabs Help
magpi@raspberrypi:~ $ sudo apt install claws-mail
[sudo] password for magpi:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
aspell aspell-en claws-mail-i18n libcompfaceg1 libetpan20 liblockfile-bin
liblockfile1 xfonts-100dpi xfonts-encodings xfonts-utils
Suggested packages:
aspell-doc spellutils claws-mail-doc claws-mail-tools
The following NEW packages will be installed:
aspell aspell-en claws-mail claws-mail-i18n libcompfaceg1 libetpan20
liblockfile-bin liblockfile1 xfonts-100dpi xfonts-encodings xfonts-utils
0 upgraded, 11 newly installed, 0 to remove and 16 not upgraded.
Need to get 8,174 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]

Figure 3: You can install a wide variety of software from the command line using the APT package manager

04 Making directories

You can also create a new directory with the `mkdir` command. For instance:

```
mkdir Test
```

You can delete a directory with the `rm` command, so long as you add the `-R` switch (short for recursive) to delete everything in it:

```
rm -R Test
```

Be careful with `rm` command as deleted files are not recoverable!

Top Tip

Case-sensitive

Note that file and directory names are case-sensitive, so entering `cd downloads` won't take you into the **Downloads** directory, and will elicit an error unless there is one called 'downloads'.

05 Installing software

While there is now an updater tool that appears in the taskbar of Raspberry Pi OS desktop when software updates are available, you can also update your software packages from the command line. To download the latest information on the configured software packages, enter:

```
sudo apt update
```

Here, we are using `apt` for the APT package manager used in Raspberry Pi OS with its `update` command. Note that we need to prefix it with `sudo`, to gain the required superuser privileges – you will be asked to enter your password.

This command is often used before upgrading all your software packages:

```
sudo apt upgrade
```

Or when installing new software. To install an example software package (that is in the APT repository), we use the `sudo apt install` command with the package name. For instance, to install the Claws Mail email client:

```
sudo apt install claws-mail
```

You will be shown information for it and asked if you want to proceed – enter **Y** to confirm (**Figure 3**). The installation process will then begin. Once it's installed, you will find Claws Mail in the desktop applications menu, under Internet. Alternatively, you can run it from the command line (although it will open in GUI window):

```
claws-mail
```

To remove a software package, you can use `sudo apt remove` followed by its name.

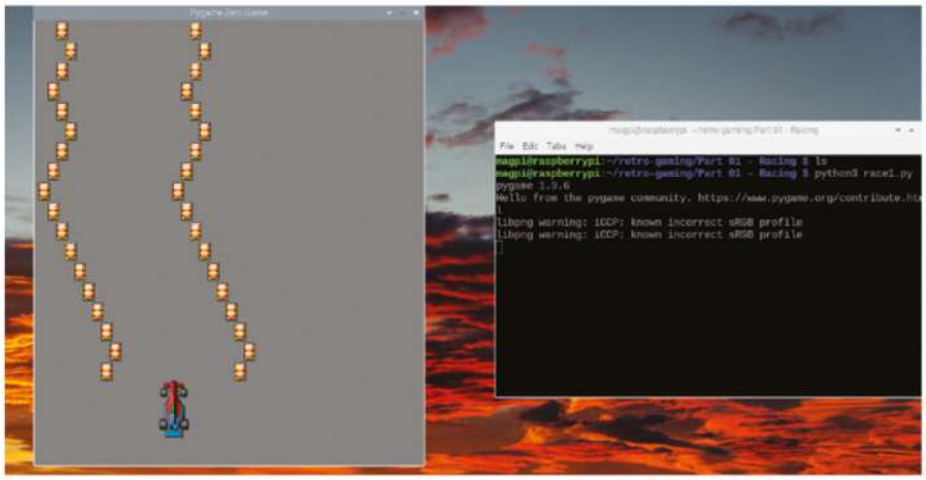


Figure 4: You can run Python programs from the command line using the `python` or `python3` command

06 Download and install

Some software may not be available from the APT repository, in which case you will need to download it (making sure it works with the ARM architecture for your Raspberry Pi) and install it. Instructions will typically be given on the website for the software. This may involve using a `curl` command to download it and run a script to install it, or a `wget` command with the URL for the file (typically a zip). You may also be asked to clone the GitHub repository for a project, such as:

```
git clone https://github.com/
themagpimag/retro-gaming.git
```

Once it has downloaded, you'll find the **retro-gaming** directory containing subdirectories with .py Python files that you can run. For instance:

```
cd /retro-gaming/'Part 01 - Racing'/
python3 race1.py
```

See **Figure 4**. To stop a program running, you can press **CTRL+C**.

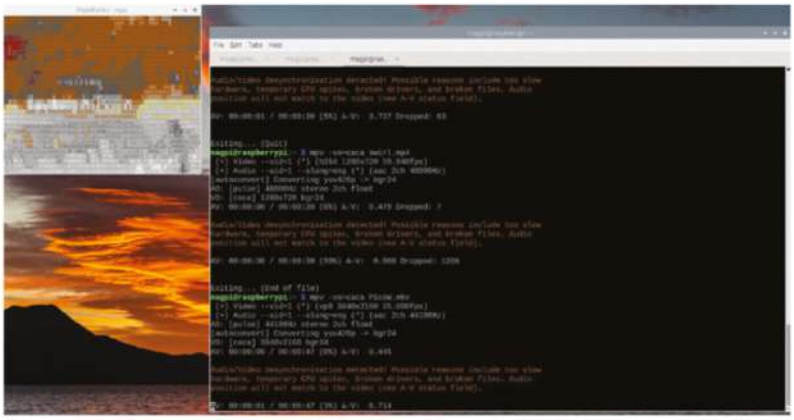


Figure 6: Playing a video in ASCII mode – can you tell what it is yet?

07 Command-line internet

You can even perform web searches directly from the command line. First, install the surfraw tool:

```
sudo apt install surfraw
```

Surfraw works with helpers called 'elvi'. To see a list of them, enter:

```
surfraw -elvi
```

Try searching for something with one, using the `sr` command.

```
sr duckduckgo Raspberry Pi
```

The results will open in a window for the default web browser, Chromium. You can, however, also view many websites in the command line itself, using the w3m text-based web browser. Install it with:

```
sudo apt install w3m
```

Then use it to view a website:

```
w3m raspberrypi.com
```

For a weather forecast, try:

```
w3m wttr.in/Poole
```

You'll see the forecast for different days, complete with ASCII graphics (**Figure 5**). Other text-friendly sites include Google.com and Wikipedia.org.

08 Audio and video

It's possible to play audio and even video from the command line using the mpv media player. Download or copy across an MP3 file and then play it with:

```
mpv example.mp3
```

You can `sudo apt install` the youtube-dl tool and then use it download a video from a link, or just use the 'v=' code at the end of the URL for it, such as:

```
youtube-dl QIdyTlmdVW8
```

This displays a Pico W video. Note that it can take quite a while to download via the proxy, however. Like any other video file, you can then use the `mpv` command with the `-vo=caca` option to play it in super-low-res ASCII (**Figure 6**)! Press **Q** to quit playing the video.

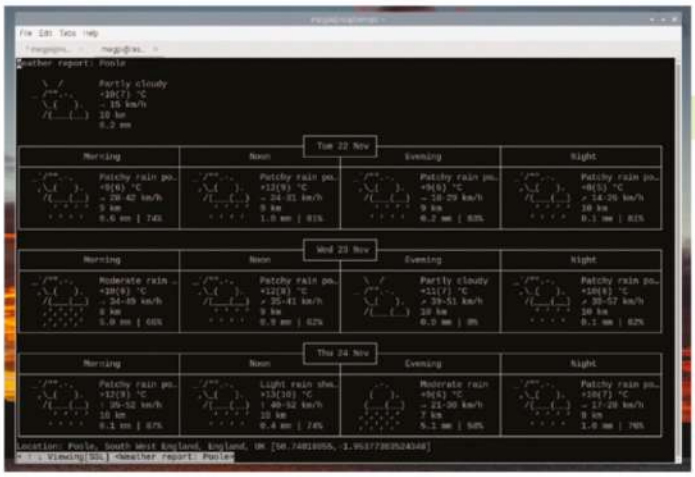


Figure 5: It's possible to browse the web from the command line. Some sites, such as wttr.in, are designed for text-based browsers

Top Tip

Software search

To search for software packages in the APT repository, use `apt-cache search` followed by the search term. Alternatively, download the Sources.gz file from **magpi.cc/aptsourcesbullseye** and use gunzip to extract it for the full list of software packages.

Build a DB9 to USB joystick converter

Connect retro joysticks to a modern PC over USB using Raspberry Pi Pico

MAKER

K.G. Orphanides

KG is a writer, developer, and software preservationist. They own one of those 1990s joysticks shaped like Giger's Alien and can confirm that it's not very comfy to game with.

magpi.cc/hauntedowlbear

You'll Need

- Arduino IDE 2.0.2 or above **arduino.cc/en/software**
- PicoGamepad library **magpi.cc/picogamepad**
- DB9 joystick INO file **magpi.cc/picoconverter**
- A DB9 joystick
- Serial port (male)

We'll be using Raspberry Pi Pico to convert input from a 9-pin DB-9 joystick, via male D-sub port, into USB input recognised by any modern PC. We refer to these 9-pin D-sub connectors as 'DB-9' – because it is the most widely used term – but they are more correctly called DE-9 connectors, which you'll also find used. It's the same physical connection as an RS232 serial port and a D-sub port, which makes for a long list of terms used interchangeably.

We'll be programming Raspberry Pi Pico using the cross-platform Arduino IDE and Arduino's Mbed OS for RP2040 boards. This gives us access to Arduino's mature and very capable HID (human interface device) handling.

The project uses code and components created by RealRobots and NickZero. This tutorial assumes that you'll be programming Pico from a Debian-based Linux system such as Raspberry Pi OS. However, all the hardware and microcontroller programming elements are entirely platform-agnostic.

01 Prepare your hardware

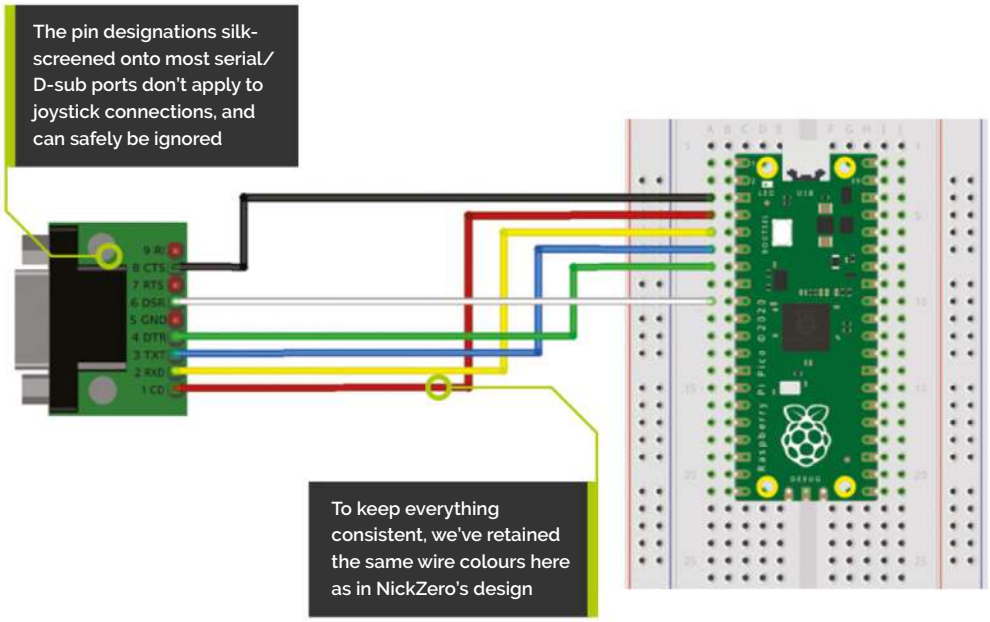
We built this project on breadboard. If you want to do the same, it's a good idea to use a serial port with a screw terminal block – both open (**magpi.cc/dbusblock**) and closed (**magpi.cc/db9block**) designs are available. Attach Raspberry Pi Pico – with headers – to a small breadboard. We like Monk Makes' Breadboard for Pico (**magpi.cc/breadboardpico**) because of its built-in GPIO cheat sheet, but any breadboard works perfectly well. See step 12 if you want to make the project a little more permanent with a case and soldered connections to a standard male D-sub port (**magpi.cc/connectorsubd**).

02 Connect the DB-9 port

You'll need six male-male DuPont jumper cables to connect the port. We don't have any specific feature requirements in our breadboard build, so you can choose whichever of Raspberry Pi Pico's GPIO pins you fancy. If you're encasing the build in any way, pick pins that won't get in your way.

As you'll see in our wiring diagram, we've connected serial port pin 1 (up) to GPIO2, serial pin 2 (down) to GPIO 3, serial pin 3 (left) to GPIO 4, serial 4 (right) to GPIO 5, serial port pin 6 (fire) to GPIO 6, and serial pin 8 (ground) to the ground connection on Pico's third physical pin. Finally, plug your joystick's connector into the port.

Our favourite joystick tester is jstest-gtk. It is available from most Linux distros' repositories


The pin designations silk-screened onto most serial/ D-sub ports don't apply to joystick connections, and can safely be ignored
To keep everything consistent, we've retained the same wire colours here as in NickZero's design

Top Tip

Get the right pinout

Different computers' joystick pinouts differ. See **magpi.cc/dejoystick** for examples and modding guides.

03 Install the Arduino IDE

Download the latest Arduino IDE from **magpi.cc/arduinoide** or, if you're running Raspberry Pi OS:

```
sudo apt install arduino
```

If you're running the IDE on Linux, including Raspberry Pi OS, you'll need to download and run as root the post-install script from **magpi.cc/postinstallsh** to create a UDEV rule allowing the IDE to write to Pico, thus:

```
sudo sh post_install.sh
```

Next, add your username to the dialout group for serial port access:

```
sudo adduser YourUsername dialout
```

04 Add the PicoGamepad library

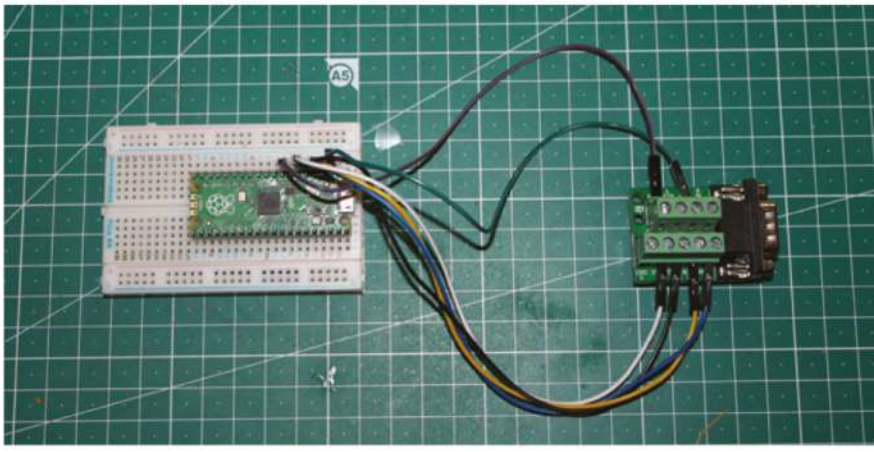
Download RealRobots' Raspberry Pi Pico Gamepad/Joystick Library as a ZIP file from **magpi.cc/picogamepad**. Open the Arduino IDE and, from the top menu bar, go to Sketch > Include Library > Add .ZIP Library, and browse to **PicoGamepad-master.zip**. To make sure that it's been added correctly, go to Files > Examples and look for PicoGamePad under Examples from Custom Libraries.

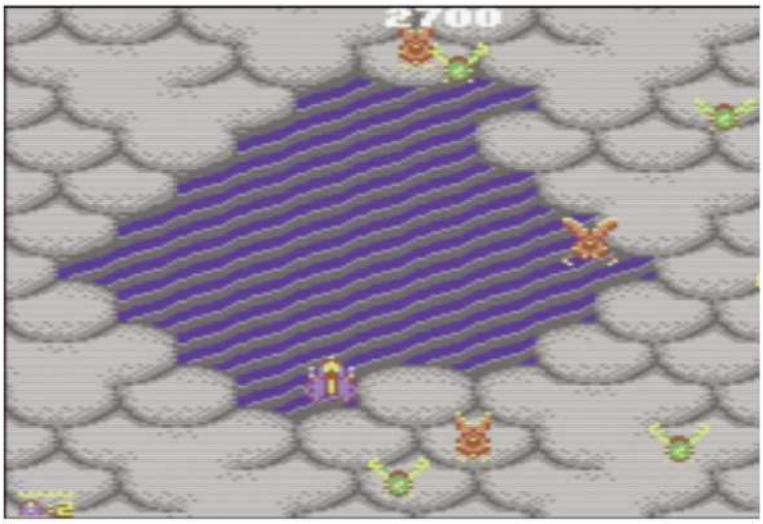
05 Customise your DB9 converter code

Download NickZero's DB9 converter code as a text file from **magpi.cc/picoconverter** or our **DB9-joystick.ino** from our repository at **magpi.cc/usbtodb9**, and open it in Arduino IDE. Where PIN_BTN_0 through 4 are defined at the beginning of a script, add the GPIO numbers for your fire, up, down, left, and right buttons. 0 is fire and the directional controls start at 1.

> Add the GPIO numbers for your buttons

We built our adapter on breadboard using a D-sub port with convenient screw terminals





Our retro joystick is ideal for playing modern C64 games like Sarah Jane Avory's Zeta Wing (**magpi.cc/zetawing**)

By default, up, down, left, and right are configured as joystick directions defined on X and Y axes. However, you can reconfigure them to fire buttons instead or as well.

06 Get ready to write

Press and hold the BOOTSEL button on Pico as you connect it to your computer. You'll have to do this every time you wish to write a sketch from the Arduino to the microcontroller. In the Arduino IDE, from the top bar, open Tools > Boards Manager and search for 'pico'. Select Arduino Mbed OS RP2040 Boards by Arduino, select the most recent version (3.3.0 at time of writing) from the pull-down, and click Install. Close Boards Manager by selecting it from from the menu again. Make sure Tools > Boards > Arduino Mbed OS RP2040 Boards > Raspberry Pi Pico is selected and that Tools > Port is /dev/ttyS0.

The joystick is immediately recognised by the Vice emulator: all you have to do is assign it. You'll want to configure it as Joystick 2 for most games, but there's a keyboard shortcut to swap them

07 Upload now

Click the upload button, a right-facing arrow, in the quick bar at the top of the IDE. If all goes well, the Output pane at the bottom of the window should show that the program has been compiled and loaded into flash.

If you want extra detail in your Output pane, open File > Preferences and select Show verbose output for compile and upload time.

If you're running Linux and getting a mysterious 'error occurred while uploading the sketch', make sure that you've run the **post_install.sh** script from step 3.

08 Test your joystick

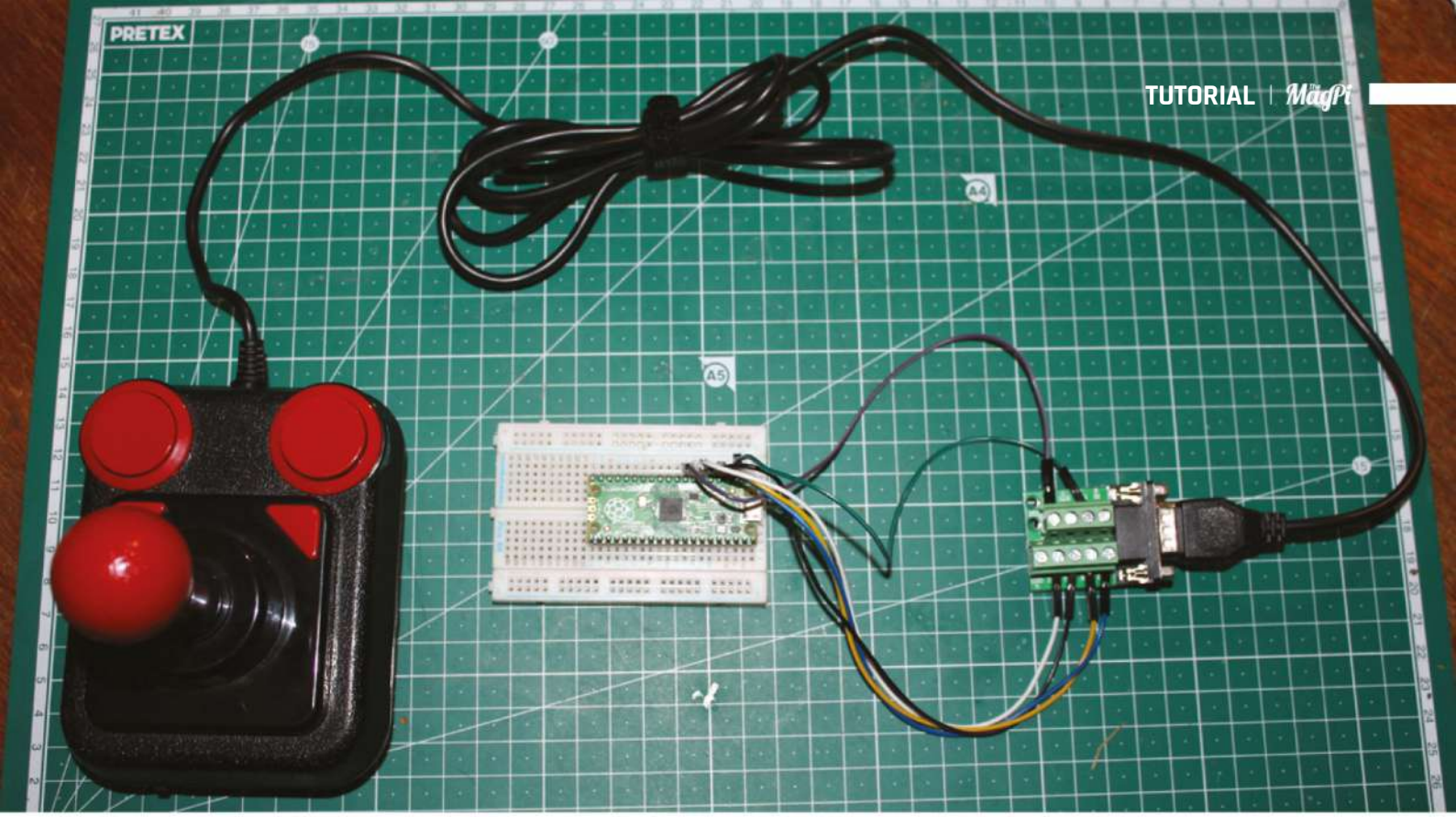
Time to install a joystick tester. On Raspberry Pi OS and other Debian-based Linux distros, open a command-line terminal and type: `sudo apt install jstest-gtk`.

Run jstest-gtk and you should see an Arduino RaspberryPi Pico associated with **/dev/input/js0** listed. Double-click this to open the button testing screen. Now test your stick and button and see what lights up. Alternatively, visit **gamepad-tester.com** for a platform-agnostic HTML5 joystick tester.

> "Test your stick and button and see what lights up"

09 Modern mappings for modern games

Now that your joystick's working, you can absolutely play Minecraft with this setup – for a given value of 'play'. We installed AntiMicroX as a Flatpak on our Linux gaming system and mapped our joystick axes to **WASD** and fire to our left mouse button. So you can move forwards, backwards, strafe, and break blocks. Turning? Who needs it? But modern games aren't really what we're going for here.



10 Retro emulation time

Install an emulator. C64 Forever and its version of Vice (**c64forever.com**) are free on Windows and you can use its licensed system ROMs with Vice on other operating systems. The Flatpak version (**magpi.cc/flathubvice**) is an easy option for Linux. You can find our guide to running Vice on Raspberry Pi in *The MagPi* 102 (**magpi.cc/102**) and find new C64 games to buy and download for free at **magpi.cc/itchc64**.

Open Vice's C64 emulator, go to Preferences > Settings > Input devices > Joystick and select the Arduino Raspberry Pi Pico. Remember, the C64's primary joystick was Joystick 2 – press **ALT+J** to swap at any time. Go to File > Smart attach, select your game's file, then its PRG in the pane on the right. Wait for it to load, then type RUN.

11 Extra buttons

If you're using a Competition Pro joystick like ours, a modern clone, or most standard joysticks from the C64 through the Amiga era, all the fire buttons are linked to pin 6, but you can physically mod your joystick to give you two separate fire buttons. Some two-button configurations may also require the addition of resistors to pins 9 (fire 2) and 7 (+5V). You'll have to modify the converter script accordingly to define and invoke fire 2, for example by creating PIN_BTN_5 and supplementing the 'Button on pin' and 'Fire Button' sections of the code accordingly. You can also find contemporary two-button joysticks, but they're relatively unusual.

12 Tidying up

NickZero's make includes STL files for a case with built-in switch for a fully soldered and encased DB9-to-USB peripheral. However, you'll probably need to iterate on the Thingiverse files provided at **magpi.cc/picoconverterfiles** to make it work for your 3D printer and components. When we had one printed by a service, we found that the hole for the D-sub port was too small for our standard male 9-pin port, while the switch designed to allow the on-board button to be pressed was too wide for the hole it's supposed to slot through. The overall design is great: the case fits together nicely and looks professional, so is well worth tweaking if you have a 3D printer and the patience to experiment.

AntiMicroX is ideal for mapping your retro joystick to whatever preposterous purpose you may put it to, from controlling your mouse cursor to (sort of) playing Minecraft

Our Competition Pro might not feel as smooth as a modern pad with analogue sticks, but it's absolutely the right tool for our retro gaming needs

Top Tip

AntiPIcrox

If you're using your joystick on a Raspberry Pi computer, AntiMicroX is ready to install from the repository with `sudo apt install antimicrox`.

Watch your fish with a Raspberry Pi camera

Remote video monitoring is easy with Raspberry Pi. Here, we use a waterproof camera case to get a unique view of some aquatic friends

MAKER

PJ Evans

PJ is a writer, software engineer and rarely without a soldering iron in-hand. He's worried his fish are watching him back.

mastodon.social/@mrpjevans

You'll Need

- Entaniya Waterproof Case for Camera Module **magpi.cc/waterproofcase**
- 100 cm cable **magpi.cc/longribboncable**
- Camera Module 2 **magpi.cc/cameramodule**

This tutorial was inspired by the Entaniya waterproof case for the Raspberry Pi Camera Module. This case protects the Camera Module without distorting the image. With it, we can submerge a camera into the depths of an aquarium and keep an eye on our fishy pals by streaming video. If that doesn't appeal, you can adapt this tutorial to be able to monitor a video stream of anything you like, whether it be indoors or outdoors. Combined with Home Assistant, you can soon be monitoring your cameras from anywhere in the world.

01 Prepare you Raspberry Pi

For the best picture quality, a Raspberry Pi 4 is the perfect choice. That said, you'll get good results from a Raspberry Pi Zero 2 W as well, and it'll be easier to install. Whichever you choose, we recommend using the Raspberry Pi Lite (Legacy) OS. The project uses the raspivid tool, support for which is limited in the latest version of the OS. The Legacy 'Buster' image still has full compatibility. You can find it in Raspberry Pi Imager under 'Raspberry Pi OS (Other)'. Set up Wi-Fi, and make sure everything is up-to-date with `sudo apt -y update && sudo apt -y upgrade.`

▲ Some assembly is required. The kit is straightforward to build and creates a watertight space for your camera

02 Install and configure your camera

With your Raspberry Pi disconnected from power, install the ribbon cable and Camera Module (we'll get to the case later). The cable will work either way around; however, the blue side (the one without the exposed contacts) needs to be against the black clip at each end. Find the connector on your Raspberry Pi marked 'CAMERA', raise the clip, gently insert the cable, then press the clip down again. Repeat for the Camera Module. Power up your Raspberry Pi and, at the command line, run `sudo raspi-config`. Go to 'Interfaces', then 'Camera', and choose 'Enable'. You'll then need to reboot.

03 Test your setup

We've set up the camera first to check everything is working before we put it in a waterproof case. After the reboot, run the following command:

The waterproof case keeps your Camera Module safe and dry without missing the action
The case floats, so some ingenuity is required to keep it in place

Top Tip

Go HQ

Although the case is for the original camera form-factor, the tutorial will work with newer HQ Cameras.

```
raspivid -f
```

All being well, you should see video from the camera on-screen for five seconds. If you don't, check if the cable is inserted correctly, and that

> You can adapt this tutorial to be able to monitor a video stream of anything you like

you've enabled the camera, as in Step 2. Now check the orientation. With the cable pointing up from the module, is the image the right way up? If not, you can switch it using this command:

```
v4l2-ctl --set-ctrl horizontal_flip=1
```

Finally, check everything is in focus and adjust the camera accordingly.

04 Install dependencies

We're going to be using a real-time video streaming server to watch our fish (or whatever you've decided). This requires some supporting software so we can build and run it, so now is the time to get everything ready. Make sure you have run `sudo apt -y update && sudo apt -y upgrade`, as in Step 1. Enter the following command:

```
sudo apt install cmake liblog4cpp5-dev
libv4l-dev git
```

This will install everything you need.

05 Build the video streaming server

Unfortunately, the streaming server is not available on APT, so we have to build it ourselves. This is straightforward, providing you enter the commands below carefully and in order. From the command line, enter the following:

```
cd
git clone https://github.com/mpromonet/
v4l2rtspserver.git
cd v4l2rtspserver
cmake .
make
sudo make install
```

This will download the source code, prepare a configuration for Raspberry Pi, and then compile the software and install it. Once installed, you

Warning!
Water

Water can damage low-voltage electrical equipment. Water with mains-voltage electricity can be deadly.

magpi.cc/electricalsafety

Top Tip

Stream better

It's good to get to know the command-line options for v4l2rtspserver as you can adjust the stream size, frame rate, and quality.

The assembled case. Now the camera is protected from the elements, whether it be in an aquarium or outdoors

can then delete the **v4l2rtspserver** directory if you wish.

06 Test your video stream

Before we start filming the fish, let's test if the video stream itself is working. On another computer, install VLC (**videolan.org**). This is a multi-purpose video playback application that supports the server we are using. From your Raspberry Pi, run this command:

```
v4l2rtspserver -W 640 -H 480 -F 15 -P 8554
/dev/video0
```

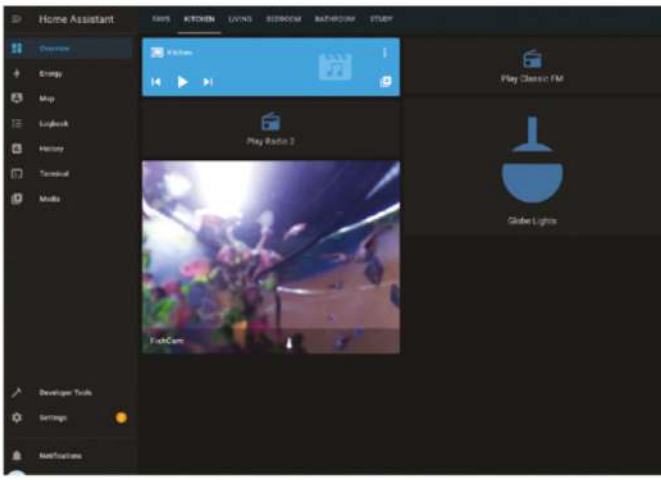
You should see some text output on the screen. On the other computer, open VLC, click on 'File' then 'Open network'. In the URL box, enter:

```
rtsp://<ip-address>:8554/unicast
```

Replace `<ip-address>` with the IP address of your Raspberry Pi (use `ip addr` to find it). Click 'Open' and wait a few seconds. You will see a stream from the camera.

07 Start at boot

Now we have a working video stream, we need to make sure it always starts at boot time.



Once your stream is running, you can add it to Home Assistant so you keep an eye on our fish anywhere

A service file already exists for the server; we just need to change one line. Open the file with:

```
sudo nano /lib/systemd/system/
v4l2rtspserver.service
```

Find the line starting with 'ExecStart' and change it so it reads as follows:

```
ExecStart=/usr/local/bin/v4l2rtspserver -W
640 -H 480 -F 15 -P 8554 /dev/video0
```

Check it's exactly as written here, then use **CTRL+X** followed by **Y** and **ENTER** to save the file and close the editor. Now enable the service:

```
sudo systemctl enable v4l2rtspserver
```

To test it, reboot, and then use VLC again to open the stream.

08 Mount the camera in the case

Let's turn our attention to the waterproof case. Following the included instructions, separate the base from the outer cover. Now open the supplied desiccant, and place both leaves in the centre of the base in the space provided. If any moisture gets in, these will absorb it, protecting your camera. Screw in the adapter mount for the camera, then carefully thread the connected ribbon cable through the slot. Finally, mount the camera to the adapter, being careful not to over-tighten the screws. Make sure the cable is fitted correctly.

09 Install the camera cover

Locate the rubber O-ring and place it on the base so it sits in the ridge provided. Take the clear cover and place it over the base, and secure it with the M3 nuts and bolts. We recommend going across each one rather than around the base; it will ensure a more even fit. Again, be careful not to over-tighten the screws. Snap the outer cover over the clear cover. Take the small rubber grommet and, very carefully, feed the ribbon cable through, threading it through the grommet until it meets the base. Insert in the slot and secure with the metal plate. You now have a waterproof case.

10 Mount your waterproof camera

What happens next will require some initiative. Depending on what it is you want to monitor, this determines how you will set things up. For our aquarium project, you can either attach some rubber 'suckers' to keep the camera in place, or find a suitable place to rest it on the substrate or sand. For outdoor use, Entaniya sells a wall-mount adapter to make things easier. Either way,

> If your interest lies more with remote monitoring, try adding presence detection or facial recognition to Home Assistant alongside the video feeds

you still need to consider how to get the ribbon cable inside or out of the tank and to a safely installed Raspberry Pi. Your biggest restriction is the length of the cable itself.

11 Add to Home Assistant

One of the best ways of utilising this monitoring solution is Home Assistant, the popular home automation operating system. Luckily, support for a remote camera like this is baked right into the service. To add the stream in, you'll need to be able to access and edit the **configuration.yaml** file. Add the following text:

This little fish came a little close for comfort. Getting a clear image may take a few goes to get the best results

```
camera:
  - platform: ffmpeg
    name: FishCam
    input: rtsp://<ip-address>:8554/unicast
```

As before, change `<ip-address>` to your Raspberry Pi's address. Now restart Home Assistant. In your dashboard you can now add a 'Picture Glance' card using the newly created 'FishCam' entity. Now your fish can be viewed wherever you can access Home Assistant.

12 Next steps

There are many ways you can build on this project. If you've installed a FishCam in your aquarium, there's lots more to do. Monitor the aquarium temperature with a 1-Wire temperature sensor and send alarms when things go awry. Or, use a flow sensor to keep an eye on the pump. If your interest lies more with remote monitoring, try adding presence detection or facial recognition to Home Assistant alongside the video feeds. Motion detection can be used to trigger recording or alerts. Outdoor cameras can be used to keep an eye on your backyard chickens or who's at the front door. As ever, it's over to you.

The author would like to acknowledge **siytek.com**, *whose excellent video streaming tutorial informed this piece.*

Top Tip

Take a different view

The case is compatible with a number of alternative lenses. Why not try a fish-eye lens for a great view of the aquarium?

How to use a Raspberry Pi in kiosk mode

Create bespoke full-screen digital signage and interactive experiences for users on Raspberry Pi with kiosk mode

MAKER

Toby Roberts

Toby used to make secret things for the police. Now he is Maker in Residence at Raspberry Pi. Laser cutting and 3D printing are his favourites. Free merch hoarder!

raspberrypi.com/tutorials

You'll Need

- Raspberry Pi (any model)
- USB-C power supply unit (PSU)
- microSD card
- HDMI cable
- Monitor, TV, or other suitable display

Kiosks are designed to offer users specific information or a specific experience, while preventing access to any other activities on the device. They are often found in airports, shops, hospitals, cafés, and museums – any location where people need easy access to information or services like timetables, waiting times, product information, directions, self check-in machines, and so on.

Kiosk mode on your Raspberry Pi allows you to boot straight into a full-screen web page or an application without using the desktop environment. It's the foundation for many different projects where you want to display information for a dedicated interaction with a user.

01 Choosing the right Raspberry Pi and accessories

Kiosk mode will work on any Raspberry Pi. For this tutorial, we'll be using a Raspberry Pi 4, so you will need a USB-C power supply and a micro-HDMI-to-standard-HDMI cable. Older models will require a micro USB power supply and a standard-HDMI-to-HDMI cable instead.

Kiosks are left powered for long periods of time, so it's important to have a good-quality power supply for your Raspberry Pi. We recommend the official USB-C and micro USB power supplies, as they are designed to provide a consistent +5.1V with captive (attached) USB cables. Don't be tempted to use an old spare phone charger and cable you may have lying around, as they may not provide sufficiently reliable power.

02 Display and setup computer

The type of display you require depends very much on your intended project. A TV or monitor with HDMI input is all that's required for this guide, but you might find you need something more specific for an interactive experience or a more compact display. For example, the Raspberry Pi Touch Display is perfect for information dashboards, such as you might use in a Raspberry Pi weather station.

For the initial microSD card setup, you'll also need another computer connected to your network. We'll refer to this as 'your usual computer', to distinguish it from the Raspberry Pi computer that you are setting up in kiosk mode. We're going to be accessing and controlling Raspberry Pi via Wi-Fi from this computer, so there is no need for an additional keyboard and mouse to be attached to your Raspberry Pi.

03 Installing Raspberry Pi OS (32-bit)

We're going to use Raspberry Pi Imager to install Raspberry Pi OS (32-bit) onto the microSD card. Raspberry Pi Imager is available for free for Windows, macOS, Ubuntu for x86, and Raspberry

11:49:25
Our Raspberry Pi displays information on this monitor that can be left in place
Raspberry Pi is set to boot up and start running automatically and is locked down to prevent unwanted hands from interfering with our display

Pi OS. Download it to your usual computer from **magpi.cc/imager**.

Connect your microSD card to your usual computer using an SD card adapter if you need one. We recommend a minimum storage size of 16GB. Open Raspberry Pi Imager, click CHOOSE OS, and select Raspberry Pi OS (32-bit), which is the smaller-sized Raspberry Pi OS that includes a desktop environment but doesn't come with unnecessary additional applications for this project.

Press **CTRL**+**SHIFT**+**X** to bring up the Raspberry Pi Imager advanced menu, or click the advanced menu cogwheel button in the bottom-right corner: Set a host name of your choice; we're using kioskpi here. Check the 'Enable SSH (Secure Shell)' box and set a username and password. You'll need these details to access your Raspberry Pi later, so be sure to make a secure note of them. Configure your wireless LAN by checking the box, and replace 'XXXXXXX' with your own SSID (network name) and password.

Click SAVE to close the advanced menu, and finally select WRITE to start the process of writing the operating system to the microSD card, which will take a few minutes to complete.

04 Starting and updating your Raspberry Pi

Insert your microSD card into your Raspberry Pi and connect it to the mains power via a USB-C power supply unit. Your Raspberry Pi will now boot for the first time and will connect to your Wi-Fi network automatically, as we have already given it all the credentials it needs.

05 Connect to your Raspberry Pi via SSH

Open a terminal session on your usual computer and run the following to access your Raspberry Pi via SSH, replacing 'pi' and 'kioskpi' with the username and hostname you chose.

```
ssh pi@kioskpi.local
```

The first time you do this, confirm that you want to connect, answer 'yes' and press **RETURN**. When asked, enter the password you created in Raspberry Pi Imager (in Step 3).

It's best practice to run an update task for a fresh installation of Raspberry Pi OS, and it's something that you should do regularly. You want to ensure the device is running with the very latest versions of all programs and security. Run the following two commands, typing **Y** and hitting **ENTER** when prompted to continue:

> "Kiosks are left powered for long periods of time"

```
sudo apt update
sudo apt full-upgrade
```

Reboot your device to allow any changes to take effect:

```
sudo reboot
```

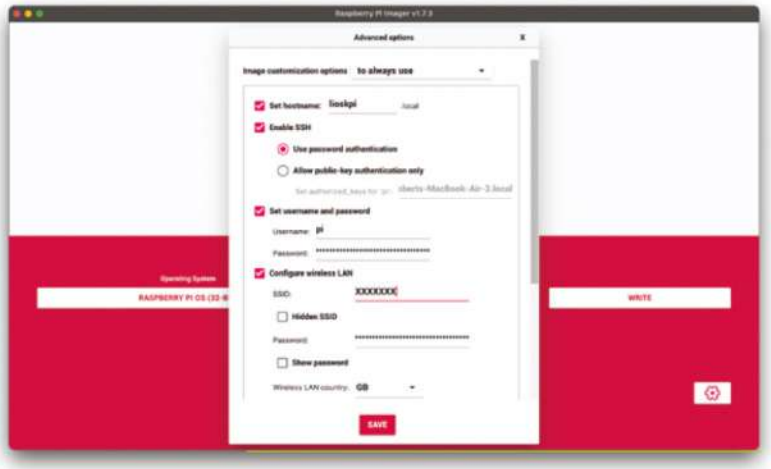
Now that your Raspberry Pi is fully updated and ready to go, log back in before moving on.

Top Tip

Stop the service

The following command will stop the service if you ever need to close the browser:

```
sudo systemctl stop kiosk.service
```



▲ Imager's Advanced Options mode is used to give our Raspberry Pi a hostname, password, and enable SSH (Secure Shell) access

06 Setting up kiosk mode

You only need two additional packages for kiosk mode to work:

- **unclutter** – removes the mouse pointer from your display
- **xdotool** – simulates keyboard and mouse activity

To install these programs, just run the following command:

```
sudo apt install xdotool unclutter
```

In order to make sure your Raspberry Pi boots straight into the desktop environment when it's powered up without the need to log in, use the software configuration tool:

```
sudo raspi-config
```

Navigate through the menu, selecting '1 System Options', then 'S5 Boot / Auto Login', and finally 'B4 Desktop Autologin – Desktop GUI, automatically logged in as 'pi' user'. Press **ENTER** and click Finish to save this setting. You will be prompted to reboot again; log back in once you have done this. If you haven't already connected an HDMI display to your Raspberry Pi, now is the time to do so. Once you have rebooted your Raspberry Pi, you should see the desktop environment on your display.

07 Create the shell script

Next, you need to create a small script to tell your Raspberry Pi what to present in kiosk mode and how to present it. For this tutorial, we want to display both the Raspberry Pi and the Greenwich Mean Time web pages, switching back and forth between the two sites. First we're going to use nano, a text editor, to create our script:

```
sudo nano /home/pi/kiosk.sh
```

Type out, or copy and paste, the **kiosk.sh** code into your file. Press **CTRL**+**X**, then **Y**, and finally **ENTER** to save the new file.

08 Create a service

In order to get Raspberry Pi to run our kiosk whenever it is powered on, we need to create a service. Use the text editor again to create another file to do this:

```
sudo nano /lib/systemd/system/kiosk.service
```

Type out the **kiosk.service** code, or copy and paste it into the file. You can see that the service starts our newly created **kiosk.sh** bash script. Hit **CTRL**+**X**, then **Y**, and **ENTER** again to save the file. Run the following command to enable our new service:

> "To run our kiosk whenever it is powered on, we need to create a service"

```
sudo systemctl enable kiosk.service
```

Either reboot or run the following to start the service:

```
sudo systemctl start kiosk.service
```

09 Adding security and fail-safe functionality

By their very nature, kiosk mode devices are often left to run unattended for long periods of time.

They are frequently turned off and on again at the mains, and left in public places, making them vulnerable to unwanted attention or interference. None of this is conducive to a reliable long-term kiosk mode installation, and kiosks can also be seen as a potential target for hackers. Whilst it's not realistic to make them 100% secure, there are a number of things we can do to protect them.

10 Disable ports

As this project currently stands, simply plugging a keyboard and mouse into the USB ports on our Raspberry Pi would give an attacker full control of our device, and potentially others on the network. Access to the Ethernet port is another weakness. The simplest way to defend against this is to disable all the unused ports. The following command will disable all USB ports:

```
echo '1-1' | sudo tee /sys/bus/usb/drivers/usb/unbind
```

To reverse this to enable the ports again, run the following:

```
echo '1-1' | sudo tee /sys/bus/usb/drivers/usb/bind
```

In order to disable the Ethernet port, enter the following command:

```
sudo ifconfig eth0 down
```

Reversing this with:

```
sudo ifconfig eth0 up
```

11 Update rc.local

These disable commands are not persistent, so when power is cycled the ports will, by default, be enabled again. The easiest way to disable them every time the device is powered on is to add to the **/etc/rc.local** file. The contents of this file are run at the end of every boot process, so it is ideal for our purpose. Edit the file using the following command:

```
sudo nano /etc/rc.local
```

Add the disable commands above the `exit 0` line and then hit **CTRL**+**X**, **Y**, and **ENTER** to save. The file should now look like this:

```
# By default this script does nothing.

echo '1-1' | sudo tee /sys/bus/usb/drivers/usb/unbind

sudo ifconfig eth0 down

exit 0
```

Each time your Raspberry Pi boots up, the USB and Ethernet ports will now be disabled.

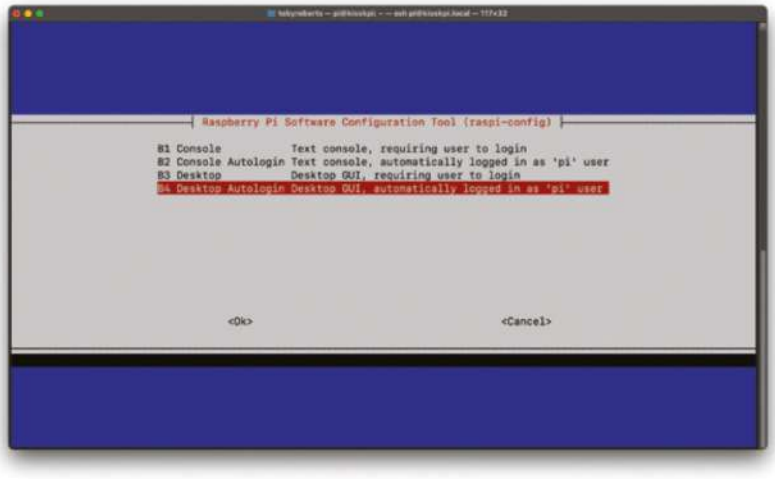
12 Turn off Bluetooth

By default, Raspberry Pi has Bluetooth turned on. For this tutorial, it is not needed, and as such it's a possible avenue of attack you can do without. To turn it off, simply edit the **/boot/config.txt** file.

```
sudo nano /boot/config.txt
```

Add the following line to the bottom of the file:

```
dtoverlay=disable-bt
```


Raspberry Pi Software Configuration Tool (raspi-config)
B1 Console Text console, requiring user to login
B2 Console Autologin Text console, automatically logged in as 'pi' user
B3 Desktop Desktop GUI, requiring user to login
B4 Desktop Autologin Desktop GUI, automatically logged in as 'pi' user
<Ok>
<Cancel>

The raspi-config tool is used to ensure Raspberry Pi boots straight into the desktop environment without the need to log in

kiosk.sh

DOWNLOAD THE FULL CODE:
magpi.cc/github

Language: **Bash**

```
001.  #!/bin/bash
002.
003.  xset s noblank
004.  xset s off
005.  xset -dpms
006.
007.  unclutter -idle 0.5 -root &
008.
009.  sed -i 's/"exited_cleanly":false/"exited_cleanly":true/'
      /home/pi/.config/chromium/Default/Preferences
010.  sed -i 's/"exit_type":"Crashed"/"exit_type":"Normal"/' /
      home/pi/.config/chromium/Default/Preferences
011.
012.  /usr/bin/chromium-browser --noerrdialogs
      --disable-infobars --kiosk https://www.raspberrypi.com/
      https://time.is/London &
013.
014.  while true; do
015.     xdotool keydown ctrl+Tab; xdotool keyup ctrl+Tab;
016.     sleep 10
017.  done
```

Press **CTRL**+**X**, **Y**, and **ENTER** to save, then reboot. Bluetooth will be disabled permanently. Simply remove the line you added to enable Bluetooth again if needed.

13 Set up SSH using keys rather than a password

Until now, connecting to our Raspberry Pi has been via SSH using a password. An added level of security is provided by connecting using pre-generated private-public RSA keys, and removing the ability to log in via username and password altogether. A private key is kept on your usual computer, the one you wish to use to connect to your kiosk, and a public key is kept locally on your Raspberry Pi. The two together provide a far better means of secure connection than a username and password. To create keys, open a terminal on your usual computer and run the following:

```
ssh-keygen
```

Accepting the defaults when prompted will create a private and a public key. The location of these saved files is also given.

The **id_rsa** key is your private key; do not share it or back it up anywhere inappropriately. It will be held locally on your usual computer. The **id_rsa.pub** is the public key, and now needs to be transferred to your Raspberry Pi. SSH back in to the device, and create the following required folder and file ready to copy and paste in the contents of the newly created **id_rsa.pub** key:

```
cd
mkdir .ssh
cd .ssh
nano authorized_keys
```

The first two lines make a normally hidden folder called **.ssh** in your home folder. Then you navigate into the new folder and create a file called **authorized_keys** using the nano text editor. Copy and paste the contents of **id_rsa.pub** from your usual computer into this file, then hit **CTRL**+**X**, **Y**, and **ENTER** to save it.

Reboot your Raspberry Pi and SSH back in. If everything works as expected, you will no longer be prompted to enter a password, and we can now disable password access completely by editing the **/etc/ssh/sshd_config** file:

```
sudo nano /etc/ssh/sshd_config
```

Find the line in the file that reads `#PasswordAuthentication yes`. Uncomment it by removing the # to enable the line, and change `yes` to `no`, so it looks like this:

```
# To disable tunneled clear text passwords,
change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

Press **CTRL**+**X**, **Y**, and **ENTER** to save the file, and you are done.

14 SD card read-only mode

Computers do not like being shut down by simply being unplugged; that is likely to result in a system file corruption at some point.

Furthermore, our operating system is stored on a microSD card, which may eventually fail if used 24/7 for long periods of time. To reduce these risks, you can make your microSD card read-only. This should be done as the final process when building in kiosk mode.

Head back into the raspi-config setting with the following command:

```
sudo raspi-config
```

Navigate through the menu system as follows:

Select '4 Performance options' > 'P3 Overlay File System'. Confirm you'd like the overlay file system to be enabled. Confirm you'd like the boot partition to be write-protected. Once done, click Finish and confirm you want to reboot.

This whole process may take a few minutes to complete.

If you need to enable access again to carry out an update or adjust your scripts, SSH back

> Computers do not like being shut down by simply being unplugged

into your device and run this command to allow access temporarily:

```
sudo mount -o remount,rw /boot
```

You can now carry out any work required. Another reboot will revert your microSD card back to a read-only state.

15 Taking kiosk mode further

This guide explores the basics of setting up a simple web page viewer in kiosk mode. Building on this might take you on a journey to more complex projects, such as a CCTV viewing station, a home automation system, or perhaps the ultimate in kiosk mode projects: a magic mirror (**magicmirror.builders**). This project was the winner in *The MagPi* magazine's 50th issue celebration feature, as voted by the Raspberry Pi community (**magpi.cc/50**).

kiosk.service

Language: **Bash**

```
001. [Unit]
002. Description=Chromium Kiosk
003. Wants=graphical.target
004. After=graphical.target
005.
006. [Service]
007. Environment=DISPLAY=:0.0
008. Environment=
     XAUTHORITY=/home/pi/.Xauthority
009. Type=simple
010. ExecStart=
     /bin/bash /home/pi/kiosk.sh
011. Restart=on-abort
012. User=pi
013. Group=pi
014.
015. [Install]
016. WantedBy=graphical.target
```

Understanding kiosk.sh

Let's break the **kiosk.sh** script down into its component parts in order to understand what it does:

- **`#!/bin/bash`** – This instructs the operating system to use bash as our command interpreter.
- **`xset s noblank, xset s off, xset -dpms`** – These commands stop the screensaver function and the desktop environment from interfering with our display.
- **`unclutter -idle 0.5 -root &`** – This stops the mouse pointer from displaying on the screen.
- The two **`sed`** lines stop any flags or pop-up warning bars, which would spoil the clean kiosk experience.
- The line beginning **`/usr/bin/chromium-browser`** specifies which websites appear in the Chromium browser in full-screen mode, and stops any user interaction with the sites. (You might not always want to prevent this, but in our example here we do, because we're making an information-only kiosk.)
- Finally, the **`while true`** lines are instructions to switch viewing between the two sites every ten seconds. During normal web browsing, the keyboard shortcut **CTRL+TAB** will cycle through the currently open browser tabs. Our script uses one of the packages we installed, xdotool, to do this by creating automatic keystrokes.

Building a web app for Pico W or Automation 2040 W

Turn almost anything into a wireless device

Ben Everard

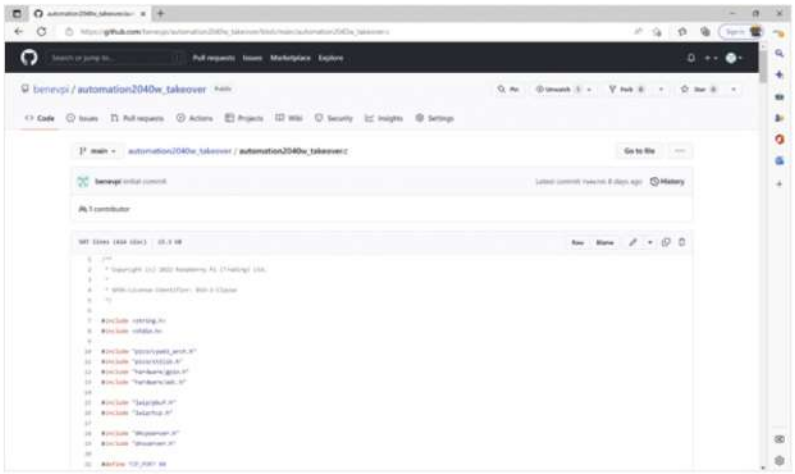
@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Below
You can grab the code from GitHub

Raspberry Pi Pico W is a fantastically powerful development board, but, like many, it's limited to a puny 3.3 V, and the GPIOs can only handle a smattering of microamps. This is fine for blinking LEDs, but if you want to run motors, or interact with more powerful components, you're going to need extra circuits. There's a whole host of options for this, including relays, motor drivers, and power MOSFETs, and you could add them to Pico W yourself. Fortunately however, the good folks at Pimoroni have already done this for you.

In this article, we'll look at working with the bare Pico 2040 W board and the Automation 2040 W. The latter beefs up most of the components: the analogue inputs, four input IOs, three output IOs, and three relays can all handle up to 40 V. In other words, it's a great device for integrating with a huge range of equipment.



The Automation 2040 W doesn't come with any firmware, as it's up to you to write code that makes sense for your project. Obviously, this could be a huge range of different things, depending on what you use the board for. To give you a sense of what's possible, we'll look at building a web interface that runs on Pico W and controls the inputs and outputs.

While there is quite a bit of circuitry on this board to handle higher voltages and currents than a bare Pico W, we control all this with GPIO pins, just as if we were controlling lower voltages. So the same technique will work on other Pico W-based boards.

In fact, we've also built a firmware for Pico W. The slight difference here is that, unlike Automation 2040 W, any pin can be either input or output. We've arbitrarily split the pins roughly in half, but you can change this so that whichever pins you like do whatever you like.

Before diving into the project, let's think a little about what our code should do:

- It should work with or without an existing WiFi network
- It should be able to read and display all the digital inputs
- It should be able to read and display all the analogue inputs
- It should be able to control all the outputs
- It should be able to control all the relays
- You should be able to control the outputs easily from a program running on another machine.

In this tutorial, we'll work backwards. To start, we'll show you how to flash the finished firmware onto your Automation 2040 W, and then we'll show you how it all works.

First, download the firmware for Automation 2040 W from **hsmag.cc/Automation2040**; or for Pico W, from **hsmag.cc/picowtakeover**. In both cases, the compiled firmware that you want to use (at least at first) is **xyz_wireless_ap.uf2**. You can flash this to your board by unplugging the USB (if connected), then holding down BOOTSEL, and plugging it back into your computer before releasing BOOTSEL.

You should then see a wireless network called **Picow_test**, and you can log into this with the password **password**. This isn't the most secure option available, and should really only be considered a test mode. You can change the SSID or password in the code and recompile if you want to.

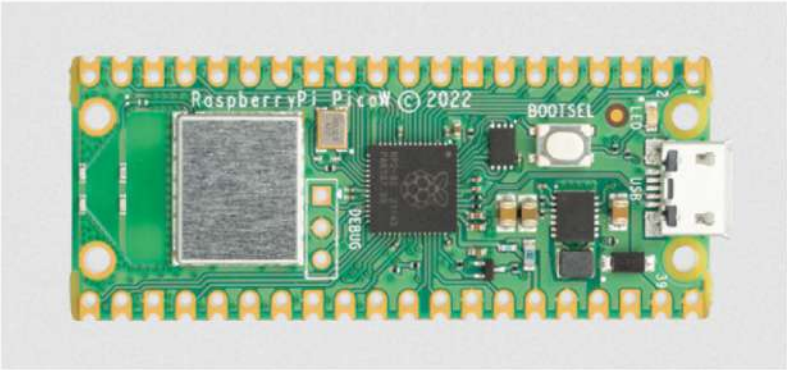
Once you've logged into that network, point your web browser to **192.168.1.4** and you'll see a web page with the basic stats of the IOs. You can see if the inputs are high or low, if the outputs are on or off, and what the voltage at the analogue inputs is.

> **The very simple web server running on Pico doesn't** really pay much attention to the URL you send to it

The very simple web server running on Pico doesn't really pay much attention to the URL you send to it. **192.168.1.4** is the IP address of the server, so you have to head there. It only listens for an HTTP request – it doesn't really care about the actual location you're requesting.

However, it does listen for parameters. In URLs, these come after a question mark, and multiple parameters are separated by an ampersand. The two parameters this server is particularly looking for are GPIO and state. Visiting the page **http://192.168.1.4/?GPIO1&state=1** will turn GPIO 1 on; visiting **http://192.168.1.4/?GPIO1&state=0** will turn it off. For general use, you don't need to know this, as there are links on the page to turn the IOs on or off. Note that the GPIO parameter refers to the actual GPIO number on Pico W rather than what is printed on Automation 2040 W.

So far, so straightforward. There are a couple of things that you might want to change: the look of the web page and the networking. To do either of these, you need to recompile the code. You can do this either by downloading and installing the toolchain, see **hsmag.cc/PicoSDKSetup** or by running it in a Google Colab environment (see box).



Above Pico W is a great device for quickly adding wireless connectivity

The CMake builds are called **automation2040w_takeover** and **picow_takeover**.

Now you've got a way of compiling the code, let's take a look at how you might want to change it. First, let's examine how it generates the HTML for the page. There's a framework for the HTML that's stored in the **macro HTML_SOURCE**:

```
#define HTML_SOURCE "<html><head><title>Automation
2040W Takeover</title></head><body><h1>Automation
2040W Takeover</h1> \
      <p>%s</p><h2>ADC</h2><p>%s</p>\
      <h2>GPIO Inputs</h2><p>%s</p>\
      <h2>Relays</h2><p>%s</p>\
      <h2>GPIO Outputs</h2><p>%s</p></body><html>"
```

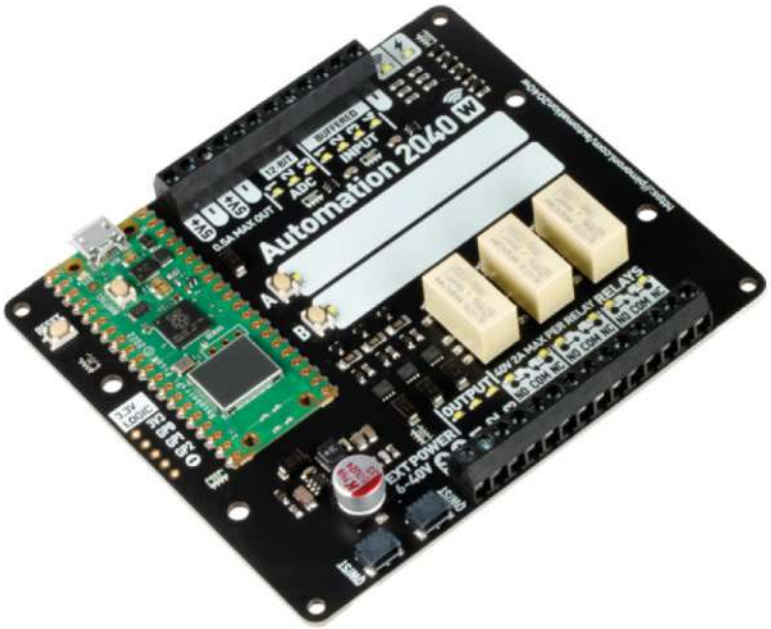
For those of you familiar with C, this should be fairly straightforward. If you're not, it might look a bit odd, but the basic thing you need to know is that macros created with a `#define` command will insert the string wherever `HTML_SOURCE` is in the code – in essence, it acts a lot like a variable (albeit a constant one). The `%s` is part of C's string formatting and identifies where we want other strings to be placed.

GOOGLE COLAB

Installing the Pico SDK on macOS or Linux is straightforward, but on Windows it's a bit of a pain. Fortunately, there is a solution – run it in the cloud. Google's Colab environment gives you an interface to an online virtual machine running Linux. You can open a page setup for this at **hsmag.cc/BuildingBlink**.

This walks you through how to compile the pico-examples, but you can use exactly the same process to compile your own code provided that you keep it on GitHub (or another online Git repository).

How long this process will work for depends on how long Google keeps Colab going. Given the firm's history of killing projects, it's possible that you might find this suddenly no longer works in the future.



Above The Automation 2040 W can take a wide range of voltages

These strings are generated a bit later in the code. First, there is a string with what (if any) GPIOs were updated during the page load:

```
int update_state = snprintf(update, 32, "<p> GPIO
%d updated to %s</p>", gpio, state?"on":"off");
```

Then we generate text for the analogue inputs:

```
#define NUM_ANALOGUE_GPIOS 3
int gpios_analogue[] = {26,27,28};

…

char gpio_analogue_lines[500] = "";
const float conversion_factor = 45.0f / (1 << 12);
counter = 1;
for(int i=0; i<NUM_ANALOGUE_GPIOS; i++) {
        if (gpios_analogue[i] == 26) {
                adc_select_input(0);
        }
        if (gpios_analogue[i] == 27) {
                adc_select_input(1);
        }
        if (gpios_analogue[i] == 28) {
                adc_select_input(2);
        }
        if (gpios_analogue[i] == 29) {
                adc_select_input(3);
        }

        uint16_t result = adc_read();

        snprintf(line, 64, "ADC %d has an
analogue voltage of %f<br>", counter,
result*conversion_factor);
        strcat(gpio_analogue_lines, line);
        counter++;

}
```

As you can see, we build up line-by-line the text that we want to go in the analogue output.

The **snprintf** function is used to insert bits of text into a string. It's based on a format string which can include format specifiers where we want to put in a bit of data. Previously we used **%s** to insert a string. Here, we're using **%d** for an integer and **%f** for a floating-point number. The full parameters are (output_variable, format string, data, …).

We use the **strcat** function to add one string to the end of another, so here **gpio_analogue_lines** is built up from all the bits of text joined together. We won't go through all the code here, as it is fairly similar, but the output GPIOs are:

```
#define NUM_OUT_GPIOS 3
int gpios_out[] = {16,17,18};

…

char gpio_out_lines[500] = "";
char line[64];
int counter = 1;
for(int i=0;i<NUM_OUT_GPIOS;i++){ // get current
GPIO values NOTE -- WORK OUT WHICH ONES are
causing a problem
        if (gpio_get(gpios_out[i])) {
                snprintf(line, 64, "Output %d
is on, turn <a href='/?GPIO=%d&state=0'>off</
a><br>", counter, gpios_out[i]);
        }
        else {
                snprintf(line, 64,"Output %d is
off, turn <a href='/?GPIO=%d&state=1'>on</
a><br>", counter, gpios_out[i]);
        }
        counter++;

        strcat(gpio_out_lines, line);
}
```

Here, you can see how we add an **<a>** tag to the line to create a link with the parameter to toggle the GPIO depending on whether it's currently on or off. All these bits are then brought together with:

```
char html_string[2000] = "";
```

HackSpace

This tutorial is from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has amazing tutorials. Find out more at **hsmag.cc**.

```
snprintf(html_string, 2000, HTML_SOURCE, update,
gpio_analogue_lines, gpio_in_lines, gpio_relay_
lines, gpio_out_lines);
```

You can easily tweak these lines to output whatever HTML you like, but you need to be aware of a couple of things. Firstly, strings in C have a fixed maximum length. This is what's in the square brackets at where the variable is defined. Strings are really just arrays of individual characters.

Secondly, there's also a maximum length for the output buffer. This is the **result** element here:

```
typedef struct TCP_CONNECT_STATE_T_ {
    struct tcp_pcb *pcb;
    int sent_len;
    char headers[128];
    char result[2048];
    int header_len;
    int result_len;
    ip_addr_t *gw;
} TCP_CONNECT_STATE_T;
```

NETWORKING

By default, the program will create its own wireless network. The code that does this is:

```
#ifdef WIFI_SSID
    printf("Connecting to WiFi...\n");
          cyw43_arch_enable_sta_mode();
    if (cyw43_arch_wifi_connect_timeout_ms(WIFI_
SSID, WIFI_PASSWORD, CYW43_AUTH_WPA2_AES_PSK,
30000)) {
        printf("failed to connect.\n");
        return 1;
    } else {
        printf("Connected.\n");
    }

        printf("Starting server at %s \n",
ip4addr_ntoa(netif_ip4_addr(netif_list)));

#else
    const char *ap_name = "picow_test";
    const char *password = "password";

    cyw43_arch_enable_ap_mode(ap_name, password,
CYW43_AUTH_WPA2_AES_PSK);

    ip4_addr_t mask;
    IP4_ADDR(&state->gw, 192, 168, 4, 1);
    IP4_ADDR(&mask, 255, 255, 255, 0);

    // Start the dhcp server
    dhcp_server_t dhcp_server;
    dhcp_server_init(&dhcp_server, &state->gw, &mask);

    // Start the dns server
    dns_server_t dns_server;
    dns_server_init(&dns_server, &state->gw);
#endif
```

As you can see, this is wrapped in a preprocessor **if** statement. For people not familiar with C, the preprocessor is a text formatter that runs before the main compiler. What this says is that if we define **WIFI_SSID**, then include the code up until the **#else**; if we don't, include the latter part.

The define of **WIFI_SSID** is done in two stages. First, we can send definitions to CMake when we create our build files with a line like:

```
cmake .. -G"NMake Makefiles" -DPICO_BOARD=PICO_W
-DWIFI_SSID=yourssid -DWIFI_PASSWORD=yourpassword
```

We can then get these defines to flow through into the C preprocessor with the following in the CMake file:

```
if ("${WIFI_SSID}" STREQUAL "")
       message("Wifi SSID not defined, so will work
in AP mode")
else()
        message("Wifi SSID defined, so will work in
station mode")
        target_compile_
definitions(automation2040w_takeover PRIVATE
        WIFI_SSID=\"${WIFI_SSID}\"
        WIFI_PASSWORD=\"${WIFI_PASSWORD}\"
        )
endif()
```

All this means that you can select your network to connect to when you run CMake, but if you don't send any details there, it will create its own access point. If you connect to an existing network, your IP address won't be **192.168.1.4** – instead, it'll be assigned by your network. If you connect to the USB serial connection, you'll see the IP; alternatively, if you can access your router, you should see the device listed there with its IP address.

Automation 2040 W and Pico W are great options for adding connectivity to non-connected devices. Here, we've shown one way to easily let you control them from your laptop or mobile phone, but there are loads of other options.

Below
Flip the Automation 2040 W over for details of how to connect it

Get Creative

WITH RASPBERRY PI PICO

Use code and components to make art, music, and much more thanks to Pico

Making isn't just about writing code and controlling robots – knitting, carpentry, painting, and much more can fall under the umbrella of making. We love to see the broad range of ways people can get creative in making, especially when a Raspberry Pi is involved.

With Pico, there's even more ways to be artistic or add some electronics to a different kind of craft. Let's see where we can squeeze one into…

pico ART

Use lights and paint with Pico to make something amazing

Conductive paint art

Using conductive paint such as the stuff from Bare Conductive (bareconductive.com), you can connect physical painted art to a Pico. LEDs can be controlled for light-up maps or drawings, and you can even try out low-power servos as well for something a little more interactive. With Pi Cap from Bare Conductive, you can also add some capacitive touch response.

There's also a load of great simple projects from Bare Conductive you can take a look at here: **magpi.cc/bcprojects** – although they'll need a little translating to work with Pico properly.

magpi.cc/conductiveart

Fiber Optic Matrix Display

Using LEDs such as NeoPixels is a pretty easy thing to do – we've covered them extensively in issues 122 and 123 with light-up vacuum cleaners and Christmas tree lights. This project (which we covered in issue 118: **magpi.cc/118**) goes further, using optic fibres connected to an LED board to create beautiful glowing effects, as text and images are pushed through them to a 'screen'. It's mesmerising in motion, and is truly a conversation starter. It's a little time-consuming though – there are about 250 optic fibres hooked between the LEDs, and the display and programming them is a little different than a standard uniform display. At least you'll have a head start on the code from maker elliotmade.

magpi.cc/fomatrix

Raspberry Pi Project

Light painting

Light painting is a bit of a trick – you're not actually painting with light, you're just using maths and a long exposure time to create something that only exists in camera. However, it is an exceptionally cool existence. You can do it with bikes with lit-up spokes, special LED strips, and much more to make incredible stuff. Check out one way here: **magpi.cc/lightpainting**.

wearing Pico

Small and powerful, wearables are now better than ever with Pico

Cyber Glasses

These glasses not only use electronics and Pico to power them, they're also 3D-printed as well, making them the most hackable wearable glasses. The maker, Kevin McAleer, likes to use them to add some extra light to anything he's soldering. As the ring is mounted on a servo, it can move out of the way when it's not needed.

Each side has a servo attached and you could add whatever you want to either side – Kevin has thought about a fan to blow away soldering fumes, and we reckon a magnifying glass could also be cool.

magpi.cc/cyberglasses

Raspberry Pi Beret

It's impossible to miss Sean O'Steen when he's wearing his very colourful beret – in fact, that's the point of it! At its most basic, this beret has LEDs around the brim powered by a Pico W running through a series of preprogrammed animations for the individual lights. However, it's also network-enabled, allowing you to connect to it with your phone.

Sean uses Pico W's networking functionality to use the beret as a signalling system, letting people nearby know in a busy convention hall if he's going to turn right or left, or even just stop.

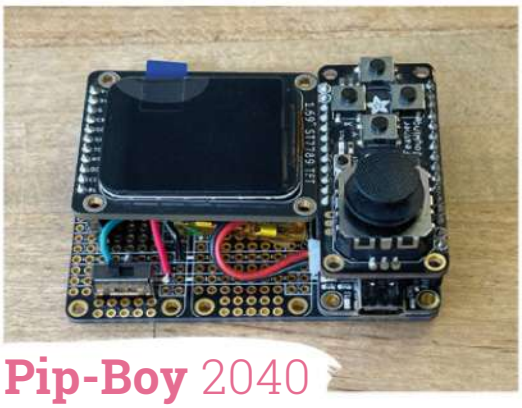
magpi.cc/picoberet

Conductive thread

When sewing electronics into a costume, you can make use of conductive thread to connect electronics to Pico, or other controller, much more discreetly. They're safe to wear and can be sometimes easier to connect than some wires – depending on the project. You can grab some at Pimoroni here: **magpi.cc/conductthread**.

Smart badges

Badger 2040 is a great little screen that easily works as a badge once connected to clothing – we've used it at cons ourselves in the past. It uses an e-ink display, so it doesn't need to be constantly connected to power to display your name, but you'll need to quickly plug it in to change to another preprogrammed screen. Grab them here: **magpi.cc/badger2040**.



Pip-Boy 2040

A classic wearable project updated for Pico. This Pip-Boy 2040 project was made by our friends at Adafruit and uses a Feather RP2040, a variant of Pico that is much smaller but still includes a full screen and usable buttons.

The code for this is a simple slide show; however, it is very easy to add more interactive screen modes by tweaking it in Python. The most difficult part is building the physical thing, and this tutorial has you well-covered there.

magpi.cc/pipboy2040

"The most difficult part is building the physical thing"

Raspberry Pi Project

Game Boy Color Costume

This is not the first time we've seen a wearable giant Game Boy powered by Raspberry Pi, and we doubt it will be the last because it is a very fun build. You could even turn yourself into a Game Gear or a Super Nintendo with an almost identical build. Check out one recent build here: **magpi.cc/gbcosplay**.

Make SOUND with Pico

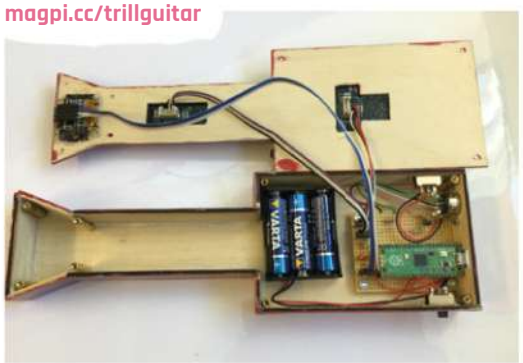
Despite not having an audio jack, you can do a lot of musical things with Pico

MIDI Guitar

Built using Trill touch sensors, this MIDI Guitar was a tutorial series in *The MagPi* a couple of years ago, starting in issue 102 (magpi.cc/102). It works much like a guitar, with 'frets' at different spots up the neck, and a touch sensor for strumming. You can change the tuning as well, which is shown on a little screen on the headstock.

The body itself is a fun bit of woodworking too, although you can always transpose the system into a different body entirely – maybe even into a wearable, musical sleeve?

magpi.cc/trillguitar



Pico Portal Music Box

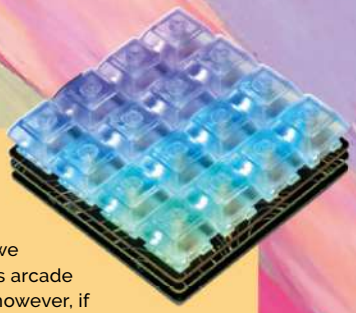
Even after 15 years, Portal's *Still Alive* song is still well remembered and this project will make sure you never forget it. Pico can actually play MP3 files, and maker thisoldgeek wanted to give it a more antique music box vibe, so he created one in the style of a music box MIDI sound. It's still an MP3 though!

He reckons it takes a couple of hours to put together, with a simple 3D-printed box and minimal amount of soldering and wiring required. It even includes lights, which fade in and out for a great effect.

magpi.cc/portalbox

Pico Audio Pack

Instead of soldering a speaker to your Pico, this add-on allows you to have high-quality audio output from Pico through 3.5 mm jacks. It connects to Pico like a HAT on Raspberry Pi. You can even choose to manually amplify each output. Grab one here: **magpi.cc/audiopack**.



MIDI controller

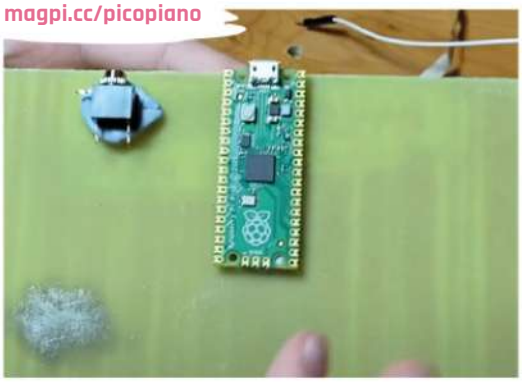
There are a couple of great DIY MIDI controller projects for Pico – we quite like the MIDI Fighter that uses arcade buttons (**magpi.cc/midifighter**) – however, if you just want to experiment with code and don't want to do too much physical building, Keybow 2040 from Pimoroni is a great customisable keyboard that can be used as a MIDI controller with some coding. Get it here: **magpi.cc/keybow2040**.

Pico Piano

This clever project uses a copper-plated board with piano keys etched into them and a series of resistors to create variable currents at each key. The piano player then uses a jumper cable hooked up to a Pico which senses the different current values and plays a note based on it.

This use of resistors would be very difficult to do on Raspberry Pi without other components. If you don't have access to copper boards and etching solutions, you can use a perf board as well to create a piano.

magpi.cc/picopiano



"The piano player then uses a jumper cable hooked up to a Pico"

Raspberry Pi Project

Lo-Fi Orchestra

This massive orchestra is made up of various microcontrollers and microcomputers, and nearly every week we get to hear something new from it. We quite like the version of Gustav Holst's *The Planets* suite composed on it. You can find out how it works with the very thorough blog here: **magpi.cc/lofiorchestra**.

YOUR GIFT GUIDE FOR 2023

THE BEST NEW PRODUCTS YOU CAN BUY
FOR CHRISTMAS AND INTO 2023

£420

G-MASTER GB3467WQSU-B1

Immerse yourself in the game with the curved GB3467WQSU Red Eagle with FreeSync Premium.

Inspired by the curve of the human eye, the 1500R curved VA panel with 165Hz refresh rate, 0.4ms MPRT, and 3440×1440 resolution guarantees superb image quality and a comfortable and very realistic viewing experience.

The height-adjustable stand ensures total flexibility for your perfect screen position. Customise the screen settings using the predefined and custom gaming modes along with the Black Tuner function to give you total control over the dark scenes and make sure details are always clearly visible.

magpi.cc/gmaster

FLIRC SKIP 1S

The Skip 1s Universal Remote Control makes controlling your entertainment center a breeze. With support for three different activities that can be activated with the click of a button, a universe of device combinations is at your fingertips. Leave the simplicity to us, and we'll leave the creativity to you.

magpi.cc/skip1s

CYBERPOWER PC

A custom gaming PC from Cyberpower UK is a perfect treat this Christmas season. With hundreds of parts to choose from – including NVIDIA's GeForce RTX 4080 GPUs – its cutting-edge configurator tool lets you build your dream spec from scratch and finish it with a custom engraved side panel.

Order your gaming PC by 12 December, or pick up a pre-built one by 21 December, for guaranteed Christmas delivery, free UK delivery, and an extended return warranty for all holiday shoppers.

cyberpowersystem.co.uk

AOC C27G2ZU/BK

Unleash your potential with this 240MHz, 27-inch display. Its 240Hz refresh rate, 0.5ms response time, and low input lag enable the AOC C27G2ZU to provide a perfectly smooth performance. With its curved design, height adjustment, and swivel ability, the monitor can be adjusted to individual needs. It comes with FreeSync Premium and G-Sync compatibility.

The C27G2ZU/BK is on a Black Friday promotion at £189, which may continue depending on stock, so pick one up early if you find it on sale.

magpi.cc/C27G2ZU

AOC Q27G2E/BK

The AOC Q27G2E/BK offers a 27-inch VA panel with QHD resolution, ShadowControl and super contrast ratio of 3000:1. Enjoy the most responsive gameplay and fastest battles with its stutter-free Adaptive Sync, 155Hz refresh rate, 1ms MPRT, and low input lag to decrease motion blur and input-output delay. Gaming has never been so fun and intense.

magpi.cc/Q27G2E

MINI PUPPER

Mini Pupper is an AI-powered, smart, quadruped robot designed for education. Mini Pupper makes robotics easier for schools, home-school families, enthusiasts, and more.

Utilising ROS (Robot Operating System) and with support for OpenCV and OpenCV's official OAK-D-Lite 3D camera module, this open-source robot is powerful and super-expandable.

mangdang.store

* Prices may vary

reTerminal E10-1

SPECS

CONNECTIVITY:
RS-485, RS-232, CAN, RJ45 Gigabit Ethernet, M.2, Mini-PCIe, mobile network, wireless LAN

DIMENSIONS:
140 mm × 95 mm × 30 mm

▶ Seeed studio ▶ **magpi.cc/reterminale10** ▶ £91 / $109

An expansion board for the industrial-focussed reTerminal computer brings much more functionality, as **Rob Zwetsloot** discovers

Powered by a Raspberry Pi Compute Module 4, reTerminal is a very functional and powerful industrial computer. It was missing a little bit of connectivity for some use cases though, most of which have been addressed with a pretty beefy expansion board.

We do mean beefy too – it's larger than the original reTerminal, and attaching it does increase the footprint (and weight). However, for that sacrifice you do get a barrel DC jack, a battery compartment, PoE, improved wireless LAN, mobile network support, serial ports, microphones, and speakers for any machine learning application, and the ability to add much more storage via SATA 2.0 and M.2 drives.

Safely secured

Adding storage isn't just a case of slotting in an SD card – the whole expansion kit comes apart so that you can install any drives into the very sturdy case so you're not losing any protection. The battery compartment is attached by screws

▲ The expansion port uses a 100-pin industrial high-speed interface

▲ The connected system is larger, but fits together nicely with no overlap from the front

as well, so any portable power you put in won't accidentally fall out.

Attaching the expansion to reTerminal is very easy – it slots into a port covered by a rubber foot

> The whole expansion kit comes apart so that you can install any drives into the very sturdy case

on the rear end, and is then secured with a couple of provided screws. No friction or plastic clasps are used so there's no danger of the connector wearing out, so to speak.

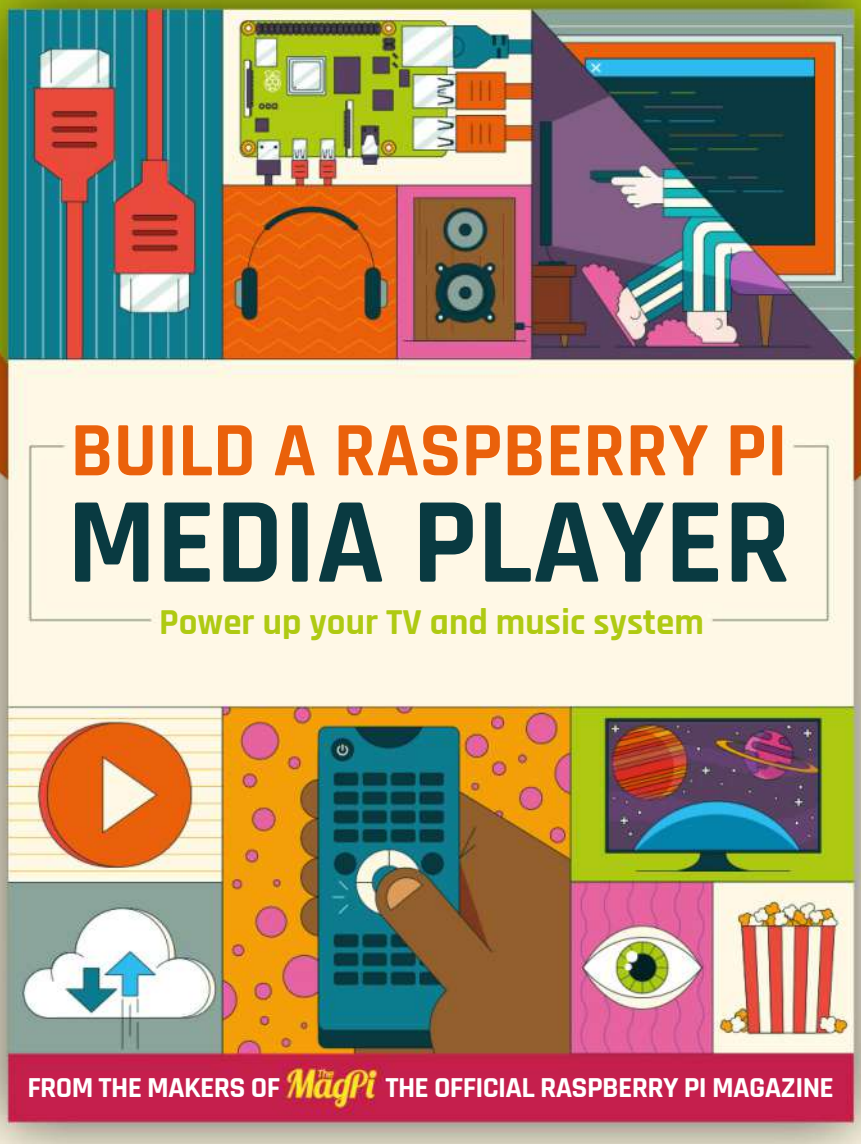
With that connection, you just need to install a driver and a Python library to the main reTerminal and you're ready to go and incorporate the expansion into the system. It's very easy and will definitely help with connecting to older hardware.

Verdict

A great, if bulky, expansion for reTerminal that seems to address any prior connectivity issues.

9/10

Your FREE guide to making a smart TV


BUILD A RASPBERRY PI
MEDIA PLAYER
Power up your TV and music system
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

magpi.cc/mediaplayer

RelayFi

SPECS

MICROCONTROLLER:
ESP32

CONNECTORS:
Relays × 4, 5 × 3 component pins, I2c breakout, USB micro data port

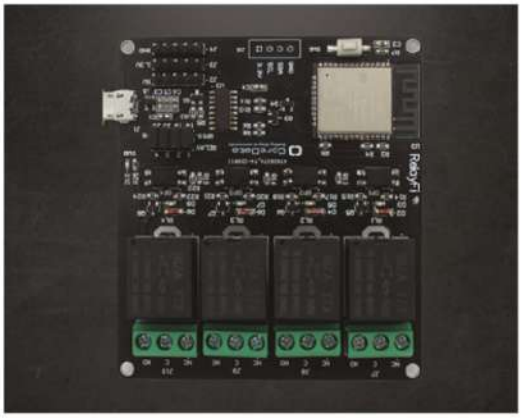
RADIO:
Wi-Fi, Bluetooth 4.2, BLE

CoreData ▶ **magpi.cc/relayfi** ▶ From £26 / $31

A 4-channel smart relay board with ESP32 and many IoT connections for a lot of home automation control. **Rob Zwetsloot** puts it to the test

ESP32 devices are a more traditional way to use IoT over a Wi-Fi network – in fact when Raspberry Pi Pico first came out, several add-on boards added Wi-Fi using it. Unlike Pico, RelayFi is much more focussed towards home automation, while also in a fairly small package.

The four relays take up most of the board and handle conversion and control for four mains-powered devices. We highly suggest getting some kind of case for the board as there are solder points on the underside of the PCB. These relays are controlled via an optoisolator which does improve the safety of the relays and, despite there being four squeezed onto the board, the screw terminals have a decent amount of space between them. Still – caution is recommended with this, like any other relay.



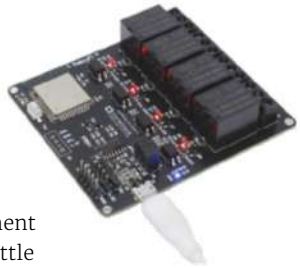
The board is fairly compact but well laid out to keep the relay side away from any parts you'll work with

Warning!
Mains Electricity

Relays can be very dangerous as they convert high currents. Only use one if you're experienced:

magpi.cc/electricalsafety

Connecting to things



USB micro is used to power and program RelayFi

With a series of I2c and standard component pins, you can add a little extra control to the board that is all programmed using the classic Arduino IDE. As it's all Wi-Fi connected, having it talk to a Raspberry Pi or Pico W is a matter of making sure it's looking for the right thing. Like an Arduino board, a micro USB port is used to power and connect to it for programming, which also supports various smart home systems for voice control via RelayFi.

> RelayFi is much more focussed towards home automation, while also in a fairly small package

It's a fairly solid build for such a relatively cheap board, and the ease of coding helps with its appeal as well.

Verdict

A solid IoT board with easy programming functions that can interact with Raspberry Pi just fine.

7/10

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE **#62**

OUT NOW

hsmag.cc

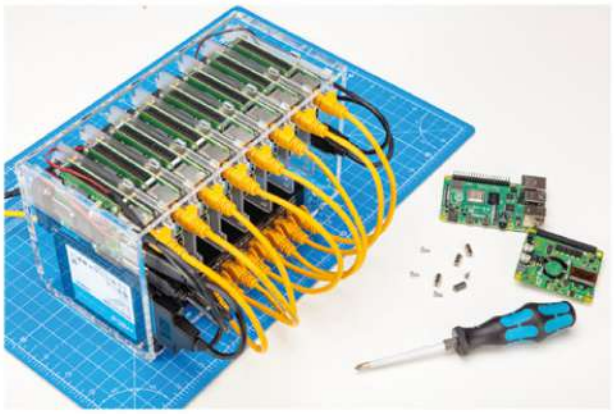
Available on the App Store

GET IT ON Google Play

10 Amazing: Advanced projects

Do more with your Pico and Pico W with these excellent add-ons

Raspberry Pi is great for people just starting out with coding and making. However, it's also used in industry for some really high-tech applications. While we're celebrating newcomers elsewhere in the mag, on these pages we have some more mature projects for Raspberry Pi veterans to get stuck in to.

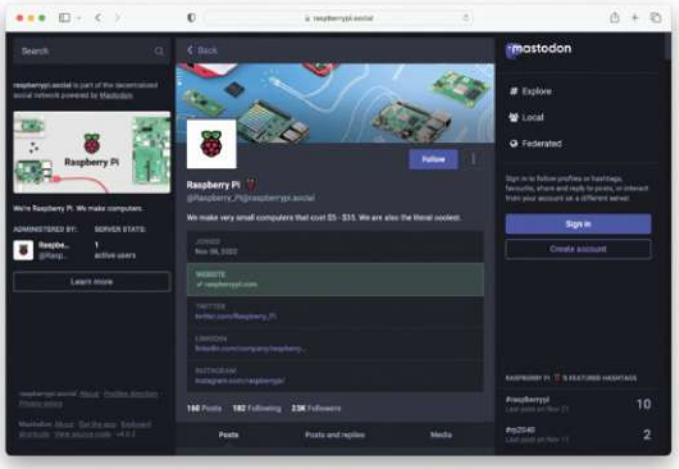


Raspberry Pi cluster

Parallel power

Turn several Raspberry Pi into your own sorta-supercomputer with some pretty decent parallel processing power – and a 1TB SSD.

magpi.cc/cluster



Host your own Mastodon instance

Social media alternative

The official Raspberry Pi Mastodon server is hosted on a Raspberry Pi server – here's how you can make one for yourself.

magpi.cc/mastodon

Web server

Online hosting

Setting up and operating a web server involves several layers of different software, including databases, admin UIs, front ends, blog themes, etc.

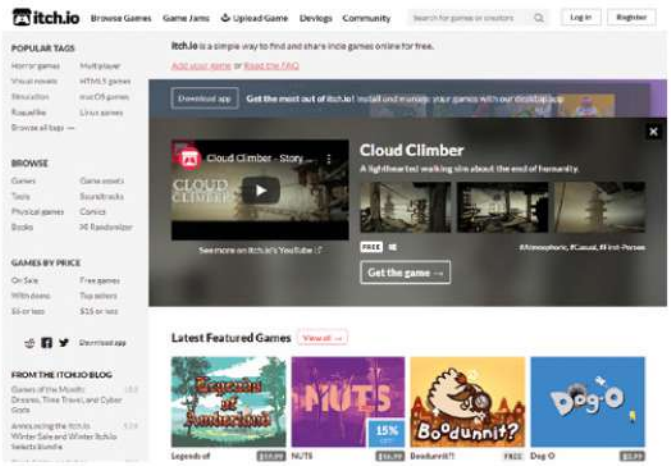
magpi.cc/webserver

OctoPrint

Social media alternative

A web interface for 3D printers powered by Raspberry Pi and full of fun plug-ins – like taking a time-lapse video so you can see your print build up (or flop) like magic.

octoprint.org



Game development

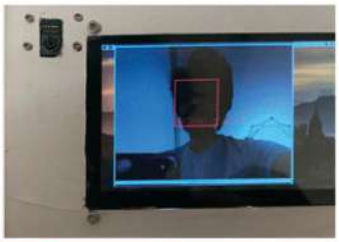
Make interactive stories

From visual novels to MicroPython platformers, there are a lot of ways to make games on Raspberry Pi, and you can even publish them on **itch.io**.

magpi.cc/gameresources

Face recognition door-lock

Automated security



This great Digi-Key tutorial uses a mixture of machine learning and actuators to make sure only the right people can enter a door. Don't run it past the Lock Picking Lawyer though.

magpi.cc/faceid

PiCar radio

Upgraded audio

Featured earlier on in this issue, this is a custom-made entertainment system for a Jeep Wrangler that aims to maintain the aesthetic of the stock radio.

magpi.cc/picar

Drone

3D-printed flyer

Raspberry Pi has gone pretty high – several are in low Earth oribt – and thanks to 3D printing and smaller electronics, it's not too tricky to make your own flying Raspberry Pi drone.

magpi.cc/builddrone

Underwater drone

Submersible photos

Waterproofing electronics can be tricky – and remote controlling them underwater is trickier still due to physics. This fantastic project manages to solve a lot of these issues.

magpi.cc/80

Pi-hole

Network security

Ads are a prime vector for malware online, and whether you have a small office or a home network to maintain, Pi-hole is a great way to keep your browsing safe.

pi-hole.net

Learn web design with Raspberry Pi

Set up your own websites with Raspberry Pi

Introduction to web development

AUTHOR

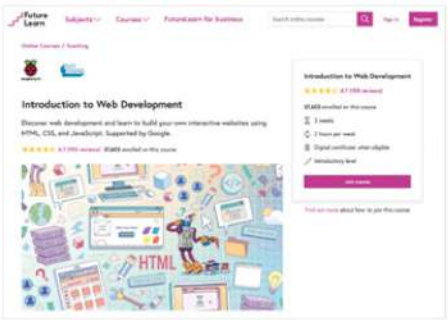
Future Learn

Price:
Free

magpi.cc/flintrowebdev

The internet remains central to our lives, and creating websites is the best way to get information across to a range of people. Plus, web design and development is a foundational skill that all coders should have.

Raspberry Pi is ideal for integrating with websites. As well as acting as a regular computer, it can be set up to act as a server to deliver web pages. And with its physical computing interface, Raspberry Pi integrates real-world items (such as sensors or motors) with online websites. This allows you to remotely control and monitor an environment via a website.



HTML, CSS, and JavaScript are cornerstone technologies for website development. Raspberry Pi's Future Learn course has all of them covered.

This free course covers HTML tabs, styling with CSS, and creating interactive elements with JavaScript. And you get a digital certificate at the end of it too.

Books

Read up on web design

HTML & CSS

Jon Duckett's twin book set for web designers and front-end developers is a lovely pair of books packed with detailed information in a clear and useful style.

▶ htmlandcssbook.com

THE PRINCIPLES OF BEAUTIFUL WEB DESIGN

If you want to make websites that don't just work, but look wonderful, this is the book to pick up. It's packed with great designs.

▶ magpi.cc/beautifulweb

LEARNING WEB DESIGN

Jennifer Niederst Robbins's book is the place to start if you are an absolute beginner. It takes you through building web pages that work using the very latest HTML5 and CSS3 code.

▶ learningwebdesign.com

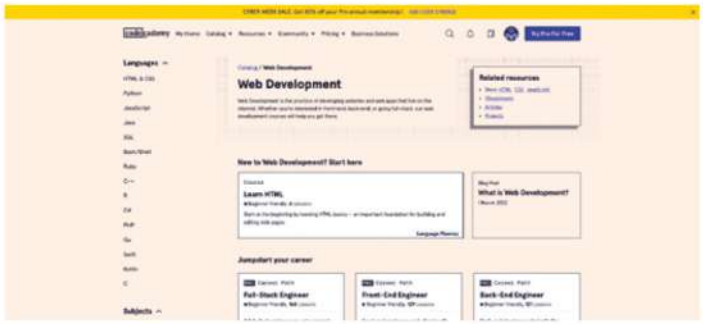
Introduction to UI & UX Design

AUTHOR

Codecademy

Price:
Free (£20/$25 Pro plan).

magpi.cc/codecademywebdev



Codecademy has a huge range of tutorials for coding and web development. What makes Codecademy special is the gamified interface that makes it fun, and rewarding, to learn. It's especially well-suited for beginners and relatively simple subjects.

All the core subjects, such as HTML, CSS, and JavaScript, are free. Plus there are some great web-integrated coding tutorials for Ruby, Node.js, Lua, and PHP. The Pro track offers up full-stack, front-end, and back-end engineering paths if you are looking to kick-start your career.

Mythic Beasts

AUTHOR

Mythic Beasts

Price:
£5/$6 (per month)

mythic-beasts.com

In the background of websites are server systems. These are the computers that serve up web pages and web-based applications.

Raspberry Pi is ideal for deployment as a web server as it's reliable, with low power requirements. Raspberry Pi can be connected to the internet and left for long periods. You can create your own web server using Raspberry Pi.

Most people hire out space on a shared web server, letting another person manage the server control. Mythic Beasts is the best of all worlds, a Raspberry Pi-based server company. You can get your own Raspberry Pi-based server, or sign up for cheaper shared hosting. It's a great way to run your own website and an extensive support system shows you everything from web and email hosting, to creating your own virtual Linux space and backup server. And because it's Raspberry Pi-based, it's cheaper, more reliable, and familiar for Raspberry Pi fans. There's a huge range of tutorials and support.

Websites

Bookmark these pages

PENGUIN TUTOR

Stewart Watkiss has put together a great page packed with information on building a website with Raspberry Pi.

magpi.cc/ptwebsite

SET UP A LAMP SERVER

Our very own tutorial shows you how to create your own LAMP (Linux, Apache, MySQL, PHP) server using Raspberry Pi.

magpi.cc/lamp

W3SCHOOLS

Packed with tutorials and reference guides, W3Schools will help you understand every single concept of web design and development.

w3schools.com

Jeff Geerling

Ex-engineer, now YouTuber who still does a lot of great engineering

> Name Jeff Geerling | > Occupation YouTuber
> Community role Maker | > URL jeffgeerling.com

Getting your kids into your interests and work is a bit hit or miss.

You can never force anyone to like anything. Luckily for Jeff Geerling's Dad – and maybe Jeff himself – he was able to share his interests.

"My Dad would bring random bits of computer hardware and electronics home from work, and he let me use his workshop to build things," says Jeff, a tech YouTuber who covers a lot of computer-based topics. "He encouraged me to try things until they worked, and instilled in me the 'hacker' spirit, where you learn by doing something wrong a few dozen times, until it works.

"And sometimes you accidentally short a computer's power supply and end up with a burn on your finger and a dead computer!"

When did you first learn about Raspberry Pi?

I remember reading about the first board on Ars Technica in early 2012. I bought my first Raspberry Pi (a Model B) on September 12, 2012, and gave it to my brother a few years later, after I started using Raspberry Pi 2 for my experiments.

How did your YouTube career get started?

I started making videos in the late 1990s, mostly for family and friends on my old Power Mac. In 2006, I started posting a few of my home videos on YouTube – a couple are still live today! I didn't dedicate much effort to YouTube until the pandemic. I had a bit of luck with a live-stream series I did called 'Ansible 101', along with some videos about building a Kubernetes cluster with Raspberry Pi computers, and my channel grew quickly.

> I learned a bit on that project, but what made it enjoyable is how much my kids loved it

I calculated what kind of viewership I would need to make producing YouTube videos a viable business plan, and once I hit those numbers, I stopped doing software consulting. I make a lot less than I used to, but it's enough. It sustains my family of six and pays the medical bills – I have Crohn's disease and it can be difficult having a chronic illness!

What are some of your favourite Raspberry Pi projects that you have made?

After the Compute Module 4 came out, I had a dream to build a one petabyte Raspberry Pi – one single Raspberry Pi with a 1PB storage volume directly attached. And I did it! That project took more than two years to pull off, and required help from Broadcom engineers, Raspberry Pi engineers, and 45Drives, a storage company. It's terribly impractical and barely worked, but it was the most fun I've had with a Raspberry Pi.

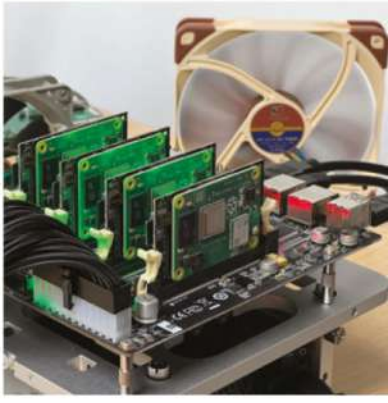
My second favourite is a slapped-together Raspberry Pi

‘Red Shirt Jeff’ detector that used AI to blast Red Shirt Jeff with my leaf blower when he came into my office. I learned a bit on that project, but what made it enjoyable is how much my kids loved it. Seeing their uncontrolled laughter when the contraption worked the first time made that project much more memorable.

What Raspberry Pi projects do your audience love?

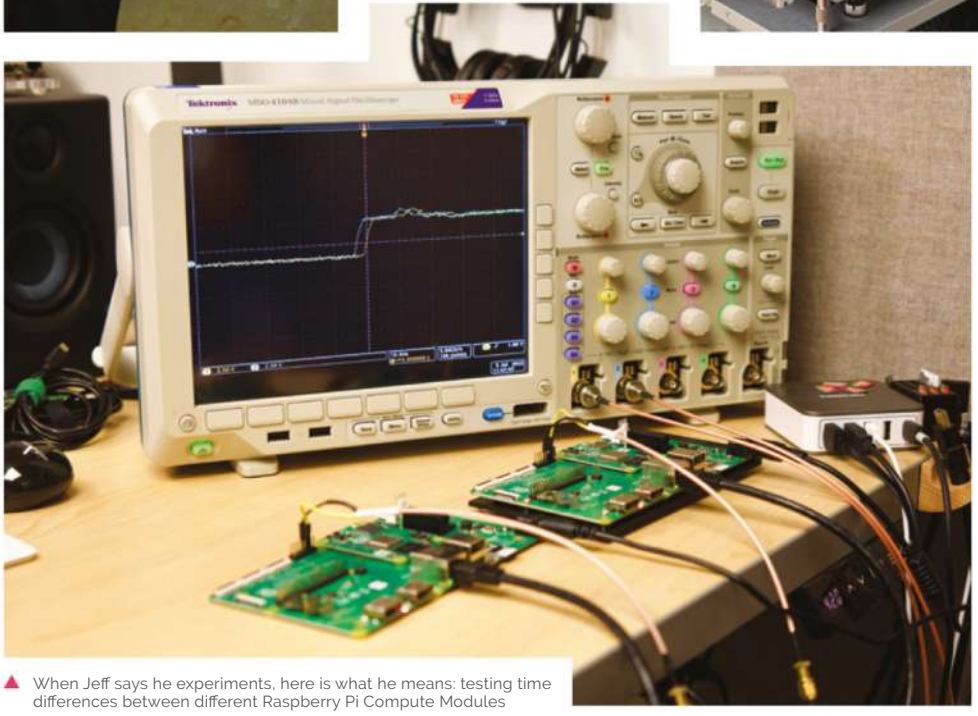
According to YouTube statistics, people love Raspberry Pi clusters, or ‘brambles’, as I call them. It started with the ‘Dramble’ (a Raspberry Pi cluster running the Drupal website **pidramble.com** since 2014), and now I’ve built a number of clusters to run Kubernetes, K3s, and even run the Linpack benchmark to see where my Pi 4 cluster ranked in the Top 500 Supercomputer list (it would’ve been competitive back in 1999)!

▲ Jeff’s website is hosted on a web server, in a barn, on a farm, powered by solar panels



◀ The petabyte cluster probably has enough data for a couple of 4K movies

▶ This is Turing Pi 2, a cluster computer made from four Raspberry Pi Compute Module 4



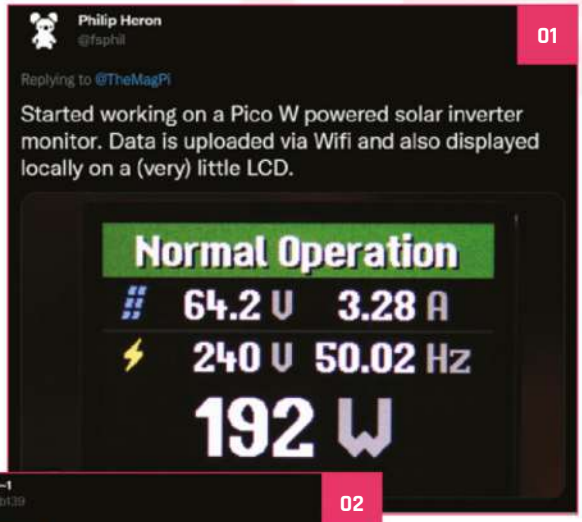
▲ When Jeff says he experiments, here is what he means: testing time differences between different Raspberry Pi Compute Modules

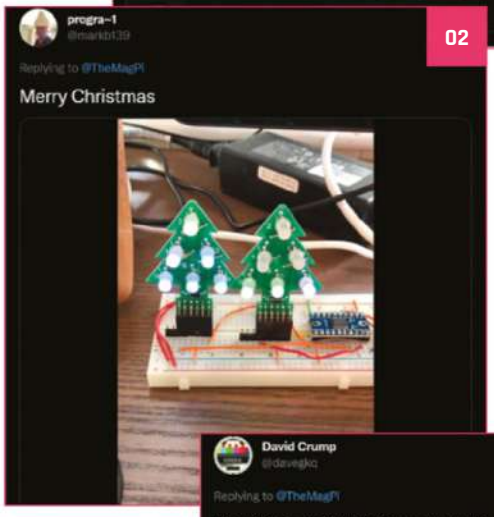
MagPi Monday

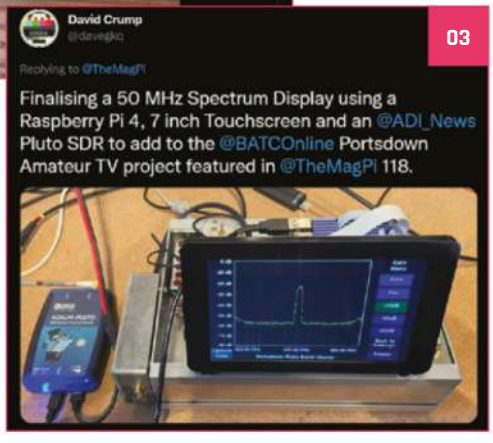
Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they have made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!


Philip Heron
@fsphil
01
Replying to @TheMagPi
Started working on a Pico W powered solar inverter monitor. Data is uploaded via Wifi and also displayed locally on a (very) little LCD.
Normal Operation
64.2 V 3.28 A
240 V 50.02 Hz
192 W




David Crump
@dave@kc
03
Replying to @TheMagPi
Finalising a 50 MHz Spectrum Display using a Raspberry Pi 4, 7 inch Touchscreen and an @ADI_News Pluto SDR to add to the @BATCOnline Portsdown Amateur TV project featured in @TheMagPi 118.

01. We like the use of a (very) little LCD to maximise the power saving of the system
02. We have a few Raspberry Pi trees we need to break out for our decorations
03. Great to see updates to projects previously featured in the magazine!
04. There's so much software that can help with stargazing on a Raspberry Pi
05. You probably don't want to roll this dice for initiative
06. This beautiful art piece is an interesting insight to modern data usage
07. The displays get better and better with each passing week
08. This PiMowBot branded remote control is adorable
09. A cheerlights-enabled bauble is our kind of Christmas decoration
10. Never be sorry about mince pies
11. A place called Punpedia sounds upsetting

04

A telescope focusser and display

Very much still in the design phase.
I found everything in my abandoned projects bin.

PS3
Stepper motor & control board
Flexi coupler
Rocker switch for focus control
HQ PiCam
3D printed gears and mounts
Display

Um, only test code so far...

05

Tweet

Colecago
@Colecago

Replying to @TheMagPi

My 55mm dice lamps use the rp2040 and circuit python: blog.colecago.com/?page_id=846

1:09 PM · Nov 21, 2022 · Twitter for Android

3 Likes

Tweet your reply

Colecago @Colecago · Nov 21
Replying to @Colecago and @TheMagPi

06

Daniel Fischer
@Artdanion

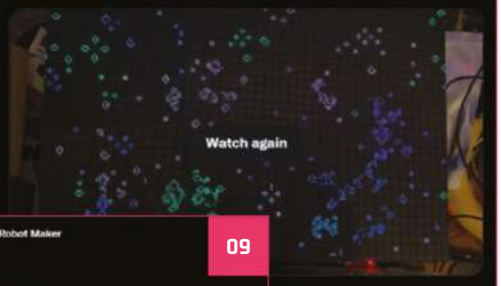
Replying to @TheMagPi

Artwork, detecting RF signals
Spectral Surrounding

07

Dr Footleg - Roboteer @drfootleg@fosstodon... @drfootl... · Nov 2
Replying to @TheMagPi

I implemented competing colours of cellular automata life in my run anywhere animations library. Running on a Raspberry Pi 4 here.
#MagPiMonday



08

TGD
@TGD Consulting

Replying to @TheMagPi

I tinkered around with the @Raspberry_Pi Pico W and a @waveshare00 IPS 1,14 display to craft a tiny RemoteControl for the PiMowBot using the PicoGraphics library for MicroPython from @pimoroni.
#MagPiMonday #PicoDisplay #RP2040 #robotics #STEM

09

Kevin McAleer Robot Maker

Replying to @TheMagPi

Happy #MagPiMonday! This weekend I built the world's most advanced / over-engineered Christmas Tree Bauble!

It has a builtin webserver and can run #cheerlights as well as other fun RGB LED Patterns, made with the @pimoroni Plasma Stick.

#micropython #robotics #STEM #adafruit

10

Brian Jamieson

Replying to @TheMagPi

I joined a cooking club.

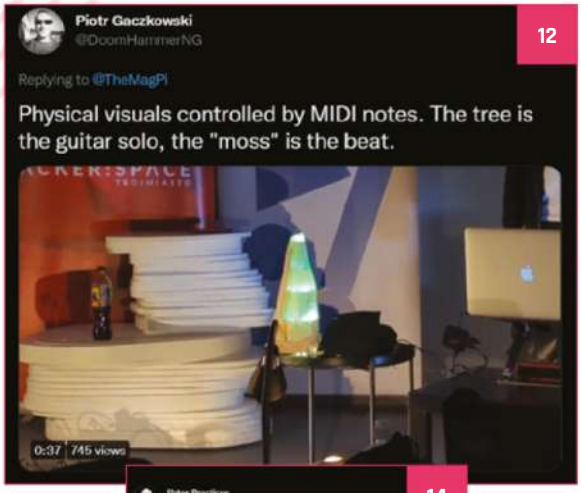
(Sorry 😉)

11

Kevin McAleer Robot Maker

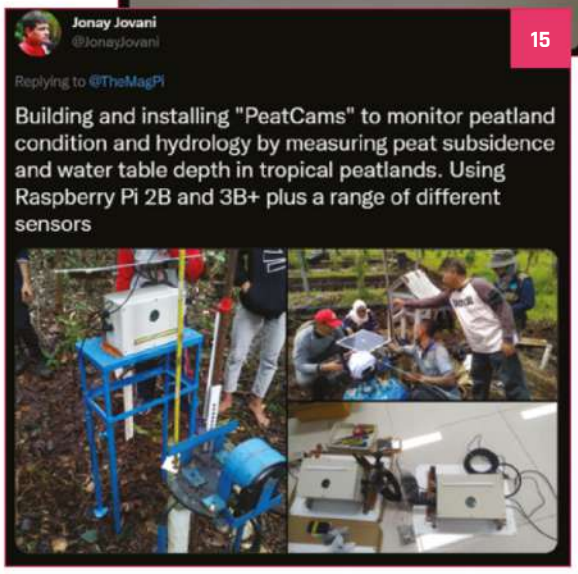
Replying to @TheMagPi

Happy #MagPiMonday!
This weekend I discovered Punpedia, which is like Wikipedia but for Puns... so I'm building an API to deliver puns to Pico W Projects, most likely the @pimoroni Inky Frame... API to be hosted on the shiny new RPi 4 that I unboxed on yesterday's Livestream.



Martin Parker
@Mr_MartinParker
13
Replying to @TheMagPi
Hey I put this up over the weekend and forgot to mention it #MagPiMonday
Martin Parker @Mr_MartinParker · 7h
No more working headless as I've got my touchscreen back on the wall since decorating, hey look it's me coming home from taking Oliver to school.

Pater Practicus
@PaterPracticus
14
Replying to @TheMagPi
Well, we got our #ChristmasTree, so I've started work on a @Raspberry_Pi Pico water monitor. Video coming soon on youtube.com/@paterpracticus


Jonay Jovani
@JonayJovani
15
Replying to @TheMagPi
Building and installing "PeatCams" to monitor peatland condition and hydrology by measuring peat subsidence and water table depth in tropical peatlands. Using Raspberry Pi 2B and 3B+ plus a range of different sensors

12. We too have always thought of trees as guitar solos
13. We like the nice wood fascia for this screen
14. Combining these two projects makes a good case for tree lights up all year
15. This is a really great bit of science

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month

PiTalk

This 4G HAT for Raspberry Pi is designed with IoT in mind. It supports a huge number of radio types, so it should work just about anywhere in the world. There's also a 2G version for slower, more low-powered applications, and a USB dongle version.

▶ **kck.st/3iek836**

PiShell 2

A trendy and protective enclosure for Raspberry Pi, or so the blurb says. We like that is has a Raspberry Pi Camera Module hole on the top of the case, and even a slot for the DSI cable so you can connect it to displays while keeping the case closed.

▶ **kck.st/3tpgA0d**

Together we can make a **difference**

Give young people the opportunity to learn about technology

The Raspberry Pi Foundation enables young people to realise their full potential through the power of digital technologies, but we can't do this work without your help. Your support helps us give young people the opportunities they need in today's world. Together we can offer thousands more young people across the globe the chance to learn to create with digital technologies.

Generous donations from organisations and individuals who share our mission make our work possible.

DONATE NOW

Donate today to make a difference: **rpf.io/magpidonate**

Raspberry Pi Foundation

Your Letters

Fish-feeding apps

Regarding the Automatic Fish Feeder Deluxe – will Erik make an iOS version of the software?

Ted via email

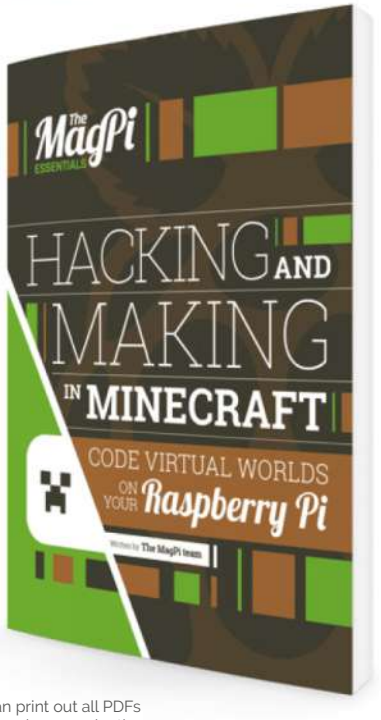
The app is fairly simple and can be used by multiple phones on the same network

The software that you is a special app for the aforementioned fish feeder from issue 124 (**magpi.cc/124**) – it's a fairly simple app that communicates with the fish feeder as long as it's connected to the same network. It uses UDP multicasting tech for this, similar to how you might stream media over your network. This means multiple phones can connect at once.

Building apps for iOS is a bit different from Android, so we're not sure how easy a port it would be. The source code is available from his GitHub (**magpi.cc/fishfeederapp**), so if you have some know-how with iOS app programming, you can probably do some porting yourself – we're reaching out to Erik anyway about it.

Contact us!

- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.com**
- Online **forums.raspberrypi.com**



You can print out all PDFs that we release under the Creative Commons licence

Print yourself

Can out of print books such as *The MagPi Essentials – Hacking and Making in Minecraft*, now only available as free PDF, be printed externally at consumer expense, or does that breach terms?

Apologies if this answer should be obvious elsewhere, thought I'd cut to the chase here.

Brian via email

Absolutely – our books and magazines are licensed under Creative Commons in a way that means you can print off the free PDFs. Not all of our books make it to print, and we can't always reprint stuff, so we're happy for you to do it – you just can't then sell the printed copies or profit from them.

Need an idea to get you started? Check out our top ten amazing Holiday projects article from issue 124

Holiday projects

I'm making something as a Christmas decoration, but I will have it finished only a few days before Christmas at the rate I'm going. Will that be too late to submit it for any kind of holiday project roundup?

Naz7 via Twitter

Due to the way making magazines works, our February issue is the one that will have any Christmas project roundups! That means that you have plenty of time to send us any holiday-themed projects that you might have before make the next magazine!

Hopefully, we'll have a few of our own to show you too! We have some examples of holiday projects in the previous issue as well, if you need some inspiration (**magpi.cc/124**).

USA SPECIAL!
6 ISSUES
FOR $42

Subscribe online:
magpi.cc/subscribe

Price is charged at £35 sterling. The dollar price will depend on the exchange rate. Six issues and free Pico W for £35 is also available in Canada and Europe. This subscription does not renew automatically. This is a limited offer. Offer subject to change or withdrawal at any time.

Wireframe

Join us as we lift the lid on video games

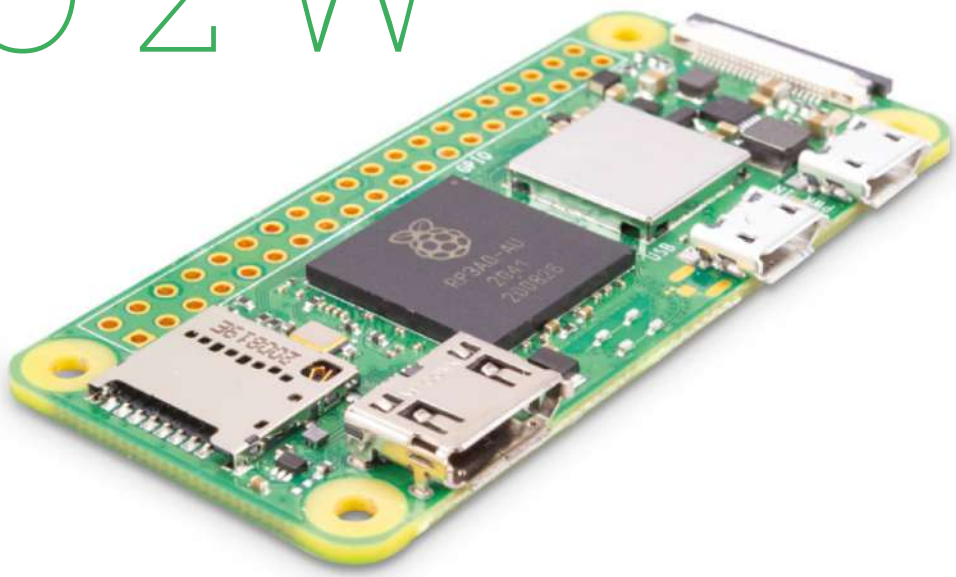
Wireframe
LIFTING THE LID ON VIDEO GAMES
PLANET OF LANA
AN OFF-EARTH ODYSSEY
LOOKING AHEAD TO THE MOST BEAUTIFUL GAME OF 2023
The secrets of Monkey Island's world-building
The surprising popularity of job simulators
Far Cry 6's level designer on telling stories via graffiti
Code your own point-and-click adventure
THE EVOLUTION OF PIXEL ART
THE HISTORY OF DEAD SPACE
ALL FORMATS

Visit **wfmag.cc** to learn more

WIN

1 OF 10 RASPBERRY PI ZERO 2 W

Start off the year with a Raspberry Pi Zero 2 W, the small yet powerful computer that fits all Raspberry Pi Zero accessories, and gives your projects a big boost in speed. We have ten to give away – enter now!



Head here to enter: magpi.cc/win | Learn more: magpi.cc/zero2w

Terms & Conditions

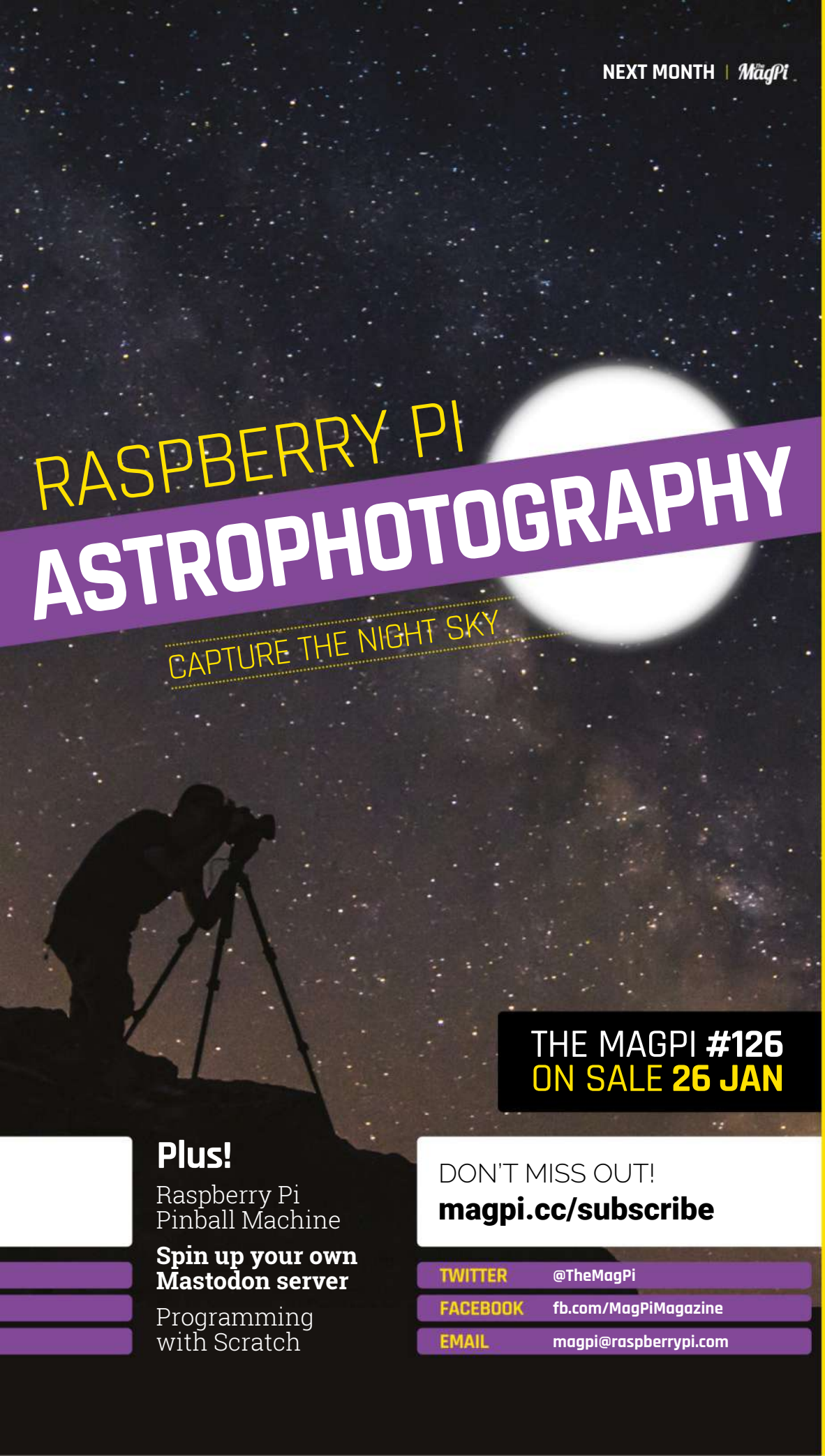
Competition opens on **14 December 2022** and closes on **26 January 2023**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter or any other companies used to promote the service.

English not your mother tongue?

The MagPi is also available in German!

Subscribe to the German edition of The MagPi and get a Raspberry Pi Pico with headers and a cool welcome box **FOR FREE!**

Use the coupon code 115PicoDE on www.magpi.de/115



THE MAGPI #126
ON SALE 26 JAN

Plus!

Raspberry Pi Pinball Machine

Spin up your own Mastodon server

Programming with Scratch

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.com

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Olivia Mitchell, Ty Logan, Sam Ribbits

Illustrator
Sam Alder

CONTRIBUTORS

David Crookes, PJ Evans, Ben Everard, Rosemary Hattersley, Nicola King, Phil King, KG Orphanides, Toby Roberts

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

recycle

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.

Achievements

Looking back at the year and planning for the next, with **Rob Zwetsloot**

In one of these columns in the past, I've definitely mentioned how I don't like to live life with regrets – if it's happened, you can't change it, you can only learn from it. I do like to remind myself of good things that have happened in the past though, especially if they inspire me to do more in the future. This also means mentally rounding up the year I've had, in the hopes that I feel like I have accomplished something.

In 2021 I struggled. I personally began streaming regularly and it turn outs that it really eats up your free time! Especially when you're starting from scratch and needing to do a crash course in OBS, overlays, and sound boards.

For 2022, I decided to make some goals and rough plans instead of stumbling into projects like I have in the past – and also promising to go a bit easier on myself if I didn't achieve them all.

Still stumbling

One thing I'm always proud of is the work we do on *The MagPi*. I'm especially happy when I get to make something cool and talk about what I've learnt in the pages of the mag – this year, in particular, three projects stand out to me.

Firstly, there were the quiz button boxes. After years of saying I'd make it, I finally created custom buzzers with programmable sounds and the ability to find out who pressed buttons in what order. I've made some modifications since I put the tutorial in the mag and, at some point, I'll update the code on the GitHub as well.

> I'm especially happy when I get to make something cool and talk about what I've learnt

Otherwise, this year I got back to making cosplay props with the Poltergust G-00 I made with help from, 3D printing wizard and Raspberry Pi Maker in Residence, Toby Roberts. It was a bit easier than last time when I had to gut a real vacuum cleaner, and I learned a lot about using NeoPixels with Pico, which I then applied to some excellent Christmas lights that are shortly going up on my tree.

Satisfaction guaranteed

Other things I planned for the year – mostly personal stuff like make silly videos and write more fiction – I wasn't always able to get to, but thankfully I did enough to be proud of. Maybe I also was able to be kinder to myself in the process as well!

Anyway, I hope you all have a fantastic 2023. I'll see you there.

AUTHOR

Rob Zwetsloot

Streaming may still be taking up a lot of Rob's time, but at least he gets to build robot models while doing it.

magpi.cc

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiShop.us PiShop.ca PiShop.mx ThePiHut Pi-Shop.ch Welectron.

Contact your favorite Pi store if it's not listed here

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!

PiKVM HAT
for DIY and custom projects

Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io